

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Дніпровський державний технічний університет
(ДДТУ)

Л. В. Дранишников

**ІНТЕЛЕКТУАЛЬНІ МЕТОДИ
В УПРАВЛІННІ**

Навчальний посібник

*Рекомендовано Міністерством освіти і науки України
як навчальний посібник для вищих навчальних закладів*

Кам'янське
"ДДТУ"
2018

УДК 519.876.5(075.8)

Д72

Рецензенти:

О. І. Міхальов — доктор технічних наук, професор, заслужений діяч науки і техніки України, завідувач кафедри інформаційних технологій та систем Національної металургійної академії України;

О. Г. Байбуз — доктор технічних наук, професор, завідувач кафедри математичного забезпечення ЕОМ Дніпровського національного університету імені Олеся Гончара;

І. М. Дорохов — доктор технічних наук, професор кафедри кібернетики РХТУ ім. Д.І. Менделєєва, президент Міжнародної академії системних досліджень, академік МАСД, заслужений діяч науки РФ

*Рекомендовано Міністерством освіти і науки України
як навчальний посібник (лист № 1/11-13114 від 13.12.2017 р.)*

Дранишников Л.В.

Д72 Інтелектуальні методи в управлінні: навчальний посібник / Л. В. Дранишников. — Кам'янське: ДДТУ, 2018. — 416 с.

ISBN 978-966-175-162-9

У навчальному посібнику розглянуті сучасні інформаційні технології: штучні нейронні мережі, нечітка логіка, генетичні алгоритми, указані шляхи використання цих технологій та наведено приклади розв'язання практичних завдань. Наводяться відомості про пакети програмного забезпечення нейронних мереж, нечіткої логіки, генетичних алгоритмів.

Навчальний посібник буде корисний для викладачів і студентів-магістрів напряму 121 спеціальності "Інженерія програмного забезпечення".

УДК 519.876.5(075.8)

ISBN 978-966-175-162-9

© Л. В. Дранишников, 2018

© ДДТУ, 2018

ЗМІСТ

ВСТУП	6
Розділ 1. ОСНОВНІ ПОЛОЖЕННЯ І МЕТОДОЛОГІЯ СИСТЕМНОГО АНАЛІЗУ	9
1.1. Системний аналіз як методологія наукового підходу	9
1.2. Методологія системного моделювання	19
1.3. Формалізація системи. Структура детерміновано-стохастичної моделі	24
1.4. Загальна схема процесу математичного моделювання. Блочний принцип побудови математичної моделі	32
1.5. Інтелектуальні методи в управлінні	44
Розділ 2. НЕЙРОННІ МЕРЕЖІ	50
2.1. Структура штучного нейрона	57
2.2. Функції активації	59
2.3. Класифікація нейронних мереж	61
2.3.1. Навчання нейронної мережі	72
2.3.2. Алгоритми навчання нейронних мереж	74
2.3.2.1. <i>Алгоритм навчання мережі за Δ-правилом</i>	82
Вправи для самостійного розв'язання	86
2.3.2.2. <i>Алгоритм зворотного поширення помилки</i>	90
Вправи для самостійного розв'язання	98
Практичні рекомендації при використанні алгоритму ЗПП...	102
2.3.3. Типи нейронних мереж	105
2.4. Пакет Neural Networks Toolbox	108
Приклади створення та використання нейронних мереж.....	108
2.5. Використання Simulink при побудові НС	111
2.6. Система автоматичного управління з нейромере- жових регулятором на основі еталонної моделі	116

2.7. Програмні пакети в області нейромережових технологій.....	121
Розділ 3. НЕЧІТКЕ МОДЕЛЮВАННЯ.....	133
3.1. Методологія нечіткого моделювання	137
3.2. Математичні основи нечітких систем	139
3.2.1. Нечіткість і вірогідність	148
3.2.2. Операції на нечітких множинах	151
3.2.3. Основні типи функцій належності	158
Задачі для самостійного розв'язання	167
3.3. Алгебра нечітких чисел. Нечіткі величини і нечіткі числа	178
3.3.1. Теоретичні основи операцій	180
3.3.2. Практична реалізація арифметичних операцій над нечіткими числами	182
3.4. Лінгвістичні змінні. Нечітка і лінгвістична змінна...	189
3.5. Нечіткі правила виводу	198
3.6. Основні етапи нечіткого виводу	204
3.6.1. Формування бази правил систем нечіткого виводу	208
3.6.2. Фазифікація.....	209
3.6.3. Агрегування	213
3.6.4. Активація	216
3.6.5. Акумуляція	218
3.6.6. Дефазифікація	221
3.7. Основні алгоритми нечіткого висновку	224
Приклад розв'язання задачі	233
Задачі для самостійного розв'язання	238
3.8. Побудова нечітких моделей у середовищі MATLAB Fuzzy Logic Toolbox	244
3.8.1. Призначення та можливості пакета Fuzzy Logic Toolbox	244
3.8.2. Графічний інтерфейс Fuzzy Logic Toolbox....	244
3.9. Нечітке моделювання при рішенні задач управління та прийняття рішень	263

3.9.1. Приклади розробки нечітких моделей прийняття рішень у середовищі MATLAB	266
3.9.1.1. <i>Нечітка модель управління кондиціонером повітря у приміщенні....</i>	266
3.9.1.2. <i>Оцінювання фінансової спроможності клієнтів при наданні банківських кредитів.....</i>	279
3.9.1.3. <i>Аналіз ризиків інформаційної безпеки</i>	296
3.10. Нечіткі нейронні мережі	308
3.10.1. Приклади розв'язання задачі нейронечіткого виводу	315
3.11. Нечіткі експертні системи	329
3.12. Програмні пакети в області нечіткої логіки	340
Розділ 4. ГЕНЕТИЧНІ АЛГОРИТМИ.....	345
4.1. Сутність еволюційних обчислень	345
4.2. Основні поняття генетичних алгоритмів	348
4.3. Генетичні оператори	360
4.4. Прийоми виконання генетичних алгоритмів	374
4.5. Програмне забезпечення генетичних алгоритмів...	387
Розділ 5. МОДЕЛЮВАННЯ СИСТЕМ АВТОМАТИЧНОГО РЕГУЛЮВАННЯ.....	390
5.1. Загальні відомості про моделювання САР	390
5.2. Моделювання типових ланок САР	394
5.3. Моделювання законів регулювання автоматичних регуляторів безперервної дії	397
5.4. Передавальні функції	400
Вправи для самостійного розв'язання.....	408
ЛІТЕРАТУРА	412

ВСТУП

Під інтелектуальними методами розуміються такі способи вирішення завдань, в основі яких лежать алгоритми і дії, в більшій чи меншій мірі пов'язані з інтелектуальною діяльністю людини. Термін "інтелектуальні" в області комп'ютерних технологій можна вважати усталеним і поєднання "інтелектуальні інформаційні технології" сприймається фахівцями однозначно.

Клас інтелектуальних технологій (ІТ) включає такі напрямки:

- штучні нейронні мережі (ШНМ);
- нечітка логіка (НЛ);
- генетичні алгоритми (ГА);
- нелінійна динаміка (НД).

Штучні нейронні мережі складаються з окремих обчислювальних елементів (формальних нейронів), які в певній мірі подібні до біологічних нейронів мозку людини. Характерна особливість ШНМ полягає в тому, що процес програмування традиційного шляхи вирішення завдання замінюється процедурою навчання мереж. Метод навчання ШНМ є одним з головних класифікаційних ознак мереж. Спосіб об'єднання нейронів в мережу, кількість шарів нейронів, наявність (відсутність) зворотних зв'язків визначають архітектуру ШНМ. Мережі різних архітектур призначені для вирішення різних завдань, у тому числі найбільш важливий клас погано формалізованих задач, де немає явних залежностей між вхідними та вихідними величинами. До проблем, які можуть бути вирішені за допомогою ШНМ, відносяться завдання класифікації і ранжирування підприємств, фірм, побудови рейтингів банків, прогнозування, зміни обмінного курсу валют і т.і.

Нечітка логіка і правила, засновані на її концепції, являють собою засіб моделювання невизначеностей природних понять мови. Прикладами таких невизначеностей служать лінгвістичні змінні холодна, тепла, гаряча стосовно температурі води. На вхід

системи з нечіткою логікою надходять чіткі змінні, які перетворюються в нечіткі підмножини. За допомогою експертів створюється база правил виду "якщо ..., то ...", що включається в систему. Над вхідними змінними за допомогою бази правил проводяться необхідні перетворення, після чого за допомогою методів дефазифікації (переходу від нечітких змінних до чітких) на виході системи формується чітка вихідна змінна. Серед управлінських завдань, що вирішуються за допомогою систем НЛ, можна виділити клас проблем, де при нечітких вхідних змінних потрібно отримати кількісну характеристику вихідної величини.

Генетичні алгоритми являють собою алгоритми пошуку оптимальних рішень, побудовані на принципах природного відбору і генетики. Будь-яке можливе рішення відображається у вигляді рядка (хромосоми) фіксованої довжини, до популяції яких застосовуються традиційні генетичні оператори: селекція, схрещування, мутація. Селекція направляє генетичний пошук у перспективні райони простору рішень, схрещування виконує випадковий пошук поблизу локального оптимуму, мутація відшукує покращене рішення. Визначаючи у в кожному поколінні кращі хромосоми, схрещуємо їх між собою для отримання потомства, яке представляє близьке до оптимального рішення задачі. До завдань, що вирішуються за допомогою ГА, можна віднести складання плану оптимальних перевезень, визначення кращої торгової стратегії, розміщення виробничих потужностей.

Нелінійна динаміка або хаос узгоджується більшою мірою щоденного досвіду людини набагато сильніше, ніж точна передбачуваність. Нелінійна динаміка - міждисциплінарна наука, в якій вивчаються властивості нелінійних динамічних систем. Нелінійна динаміка використовує для опису систем нелінійні моделі, зазвичай описувані диференціальними рівняннями і дискретними відображеннями. Методи нелінійної динаміки дозволяють моделювати швидкі нерівноважні процеси (так звані «фазові переходи») в системах, пов'язані з переходом з одних стійких станів в інші.

Сучасний фахівець, пов'язаний тією чи іншою мірою зі створенням програмного забезпечення, аналізом, прогнозуванням і прийняттям рішень, повинен володіти, поряд з традиційними знаннями, також знаннями і навичками використання останніх досягнень в області створення інформаційних систем.

У навчальному посібнику викладено математичні моделі і методи моделювання, які можуть застосовуватися для аналізу, прогнозування та прийняття рішень. Показано, що математичне моделювання є основою для економічного, кількісного і наукового розгляду технічних проблем. Наприклад, нечітке моделювання виявляється особливо корисним, коли в описі технічних систем і бізнес-процесів присутня невизначеність, яка ускладнює або навіть виключає застосування точних кількісних методів і підходів. При побудові моделей реальних явищ дослідник, бажаючи користуватися математичними методами, часто вносить визначеність там, де її немає по суті. Чим складніша система, тим глибше ми аналізуємо завдання, тим біль невизначеним стає її рішення. Л. Заде сформулював принцип несумісності: *чим складніше система, тим менше ми здатні дати точні і в той же час практичні судження про її поведінці. Для систем, складність яких перевищує деякий пороговий рівень, точність і практичний сенс стають практично виключають один одного характеристиками.*

У навчальному посібнику розглянуті сучасні інформаційні технології, засновані на деяких аналогіях з повсякденним інтелектуальною діяльністю людини, зокрема, штучні нейронні мережі, нечітка логіка, генетичні алгоритми, вказані шляхи використання цих технологій і наведені приклади розв'язання практичних завдань.

РОЗДІЛ 1

ОСНОВНІ ПОЛОЖЕННЯ І МЕТОДОЛОГІЯ СИСТЕМНОГО АНАЛІЗУ

1.1. Системний аналіз як методологія наукового підходу

Центральним поняттям системного моделювання є саме поняття система, під якою розуміється сукупність об'єктів, компонентів або елементів довільної природи, що утворюють деяку цілісність у тому чи іншому контексті. Визначальним принципом розгляду деякої сукупності об'єктів як системи є поява у неї нових властивостей, яких не мають її складові елементи. Таким чином, **система** – це сукупність елементів, яка принаймні, володіє такими п'ятьма властивостями [1,2]:

1) властивість **цілісності і членування** (система насамперед є цілісною сукупністю взаємопов'язаних елементів; вона припускає деталізацію, тобто розчленування на підсистеми; елементи системи за певних умов можуть розглядатися як самостійні системи, а сама система – як елемент більш високого рівня ієрархічного розчленування);

2) властивість **зв'язності** (система існує як цілісне утворення тоді і тільки тоді, коли сила міжелементних зв'язків усередині системи вище, ніж сила зв'язки цих елементів з довкіллям; зв'язки розрізняють за фізичним наповненням– речовинні, енергетичні, інформаційні, і за напрямом – прямі, зворотні, нейтральні; істотною характеристикою зв'язку є її сила, яка може бути оцінена кількісно). Упорядкована в просторі і часі сукупність елементів і зв'язків утворює **структуру системи**. Наочно структура системи представляється у вигляді графа, вузли якого відповідають елементам, а дуги – зв'язкам (рис.1.1);

3) **інтегративна** якість системи полягає в тому, що вона має властивість, притаманну системі в цілому, але невластиву жодно-

му її елементу окремо. Наявність інтегративних якостей показує, що властивості системи хоча і залежать від властивостей елементів, але не визначаються ними повністю. Звідси два висновки: 1) система не зводиться до простої сукупності елементів; 2) розчленовуючи систему на окремі частини, вивчаючи кожну з них окремо, неможливо пізнати всі властивості системи в цілому.

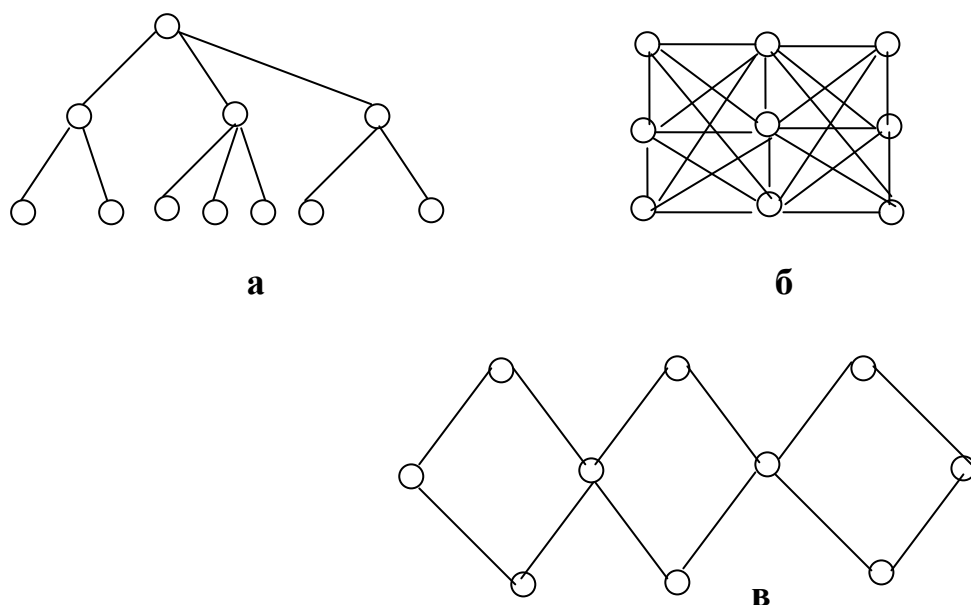


Рис. 1.1. Класифікація структур залежно від взаємодії елементів і зв'язків:
 а – ієрархічна; б – мережева; в – стільникова

4) властивість **організованості** системи полягає у наявності певної організації, що проявляється у зниженні інформаційної ентропії E (ступеня невизначеності) системи. Інформаційна ентропія є натуральний логарифм числа S допустимих станів системи, тобто $E \equiv \ln S$. Якщо система може знаходитися тільки в єдиному допустимому стані, то $E = 0$. Якщо в однієї системи число допустимих станів дорівнює S_1 а інший S_2 , то об'єднана система складається з цих двох систем, що не контактують, і буде мати $S_1 \times S_2$ допустимих станів, тоді ентропія об'єднаної системи є:

$$E_{\Sigma} = \ln(S_1 S_2) = \ln(S_1) + \ln(S_2) = E_1 + E_2.$$

Рівень організованості або упорядкованості системи оцінюється величиною:

$$R = 1 - \frac{E}{E_{\max}} = \frac{E_{\max} - E}{E_{\max}} = \frac{NE}{E_{\max}},$$

де E – поточне значення інформаційної ентропії системи; E_{\max} – максимально можлива інформаційна ентропія системи; $NE = E_{\max} - E$ – негентропія системи (накопичена інформація).

З останнього виразу видно, що перехід до більш високого рівня впорядкованості та організованості системи означає зменшення її поточної невизначеності (ентропії) за рахунок нагромадження інформації. Якщо система повністю детермінована й організована, то $E = 0$ і $R = 1$. Якщо система повністю дезорганізована, то $R = 0$ і $E = E_{\max}$, т. о. організованість системи лежить у межах $0 \leq R \leq 1$.

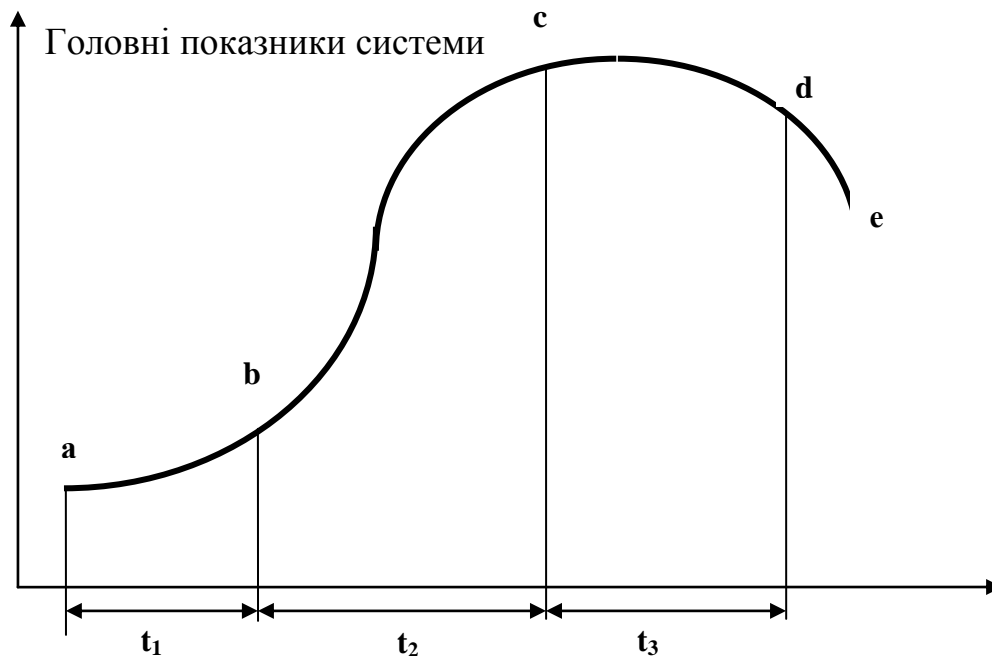


Рис.1.2. Крива життєвого циклу системи

5) властивість **життєвого** циклу системи характерна для систем будь-якого типу – природних, штучних, технічних, біоло-

гічних, соціальних тощо, й у загальному вигляді відображається S-подібною кривою (рис.1.2), що визначає характер зміни у часі головних характеристик системи (працездатність, сила, потужність, енергія, продуктивність тощо). Крива життєвого циклу має характерні для всіх систем ділянки: 1) «дитинство» (a-b); 2) «зрілість» (b-c); 3) «старість» (c-d); 4) «смерть» (d-e).

Структура і функції. Поняття структури пов'язане з другим основним поняттям системи – властивістю зв'язності. Структура системи є стійкою упорядкованою в просторі і в часі сукупністю елементів і зв'язків. Невід'ємною властивістю структури системи є її ієрархічність (співпідпорядкованість підсистем). Будь-яка система має не менше двох рівнів ієрархії: «старший» рівень – система; «молодший» рівень – елементи. Тип ієрархії структури може бути просторовим, тимчасовим, смисловим тощо.

З точки зору просторової організації розрізняють плоскі, об'ємні та довільного числа вимірів; розподілені, коли елементи розосереджені в просторі; зосереджені, коли має місце накладання елементів у локальному об'ємі (точці) простору.

З точки зору тимчасової ознаки поділяють **екстенсивні** структури, в яких з плином часу відбувається зростання числа елементів, та **інтенсивні**, в яких відбувається зростання числа зв'язків і їх потужності при незмінному складі елементів. Між ними знаходяться стабільні структури, які не змінюються протягом всього періоду життя системи.

Функція системи – вияв певних властивостей при взаємодії з зовнішнім середовищем. Відмітною властивістю системи є те, що жоден з її елементів не має притаманних їй властивостей, не може виконувати ту функцію, яку вона здійснює.

Елементи системи володіють певним набором властивостей. Одні з цих властивостей при формуванні зв'язків елементів придушуються. Інші, навпаки, проявляються сильніше. Однак ступінь придушення системою незначущих властивостей елемента, як правило, не буває повною. У зв'язку з цим при формуванні си-

стеми виникають не тільки «корисні» функції, що забезпечують збереження системою її якісної особливості, але й дисфункції, тобто функції, що негативно впливають на функціонування системи, зростання яких може привести до руйнування системи.

Вивчення будь-якої системи передбачає побудову **математичної моделі**, що відбиває певну групу властивостей системи і дозволяє прогнозувати її поведінку в певних умовах. Під моделлю будемо розуміти деяке уявлення про систему, відбиваюче найбільш істотні закономірності її структури та процесу функціонування і зафіксоване деякою мовою чи у певній формі. Модель – це математичний опис системи. Такий опис, як правило, розглядається з трьох точок зору: морфологічної, функціональної та інформаційної.

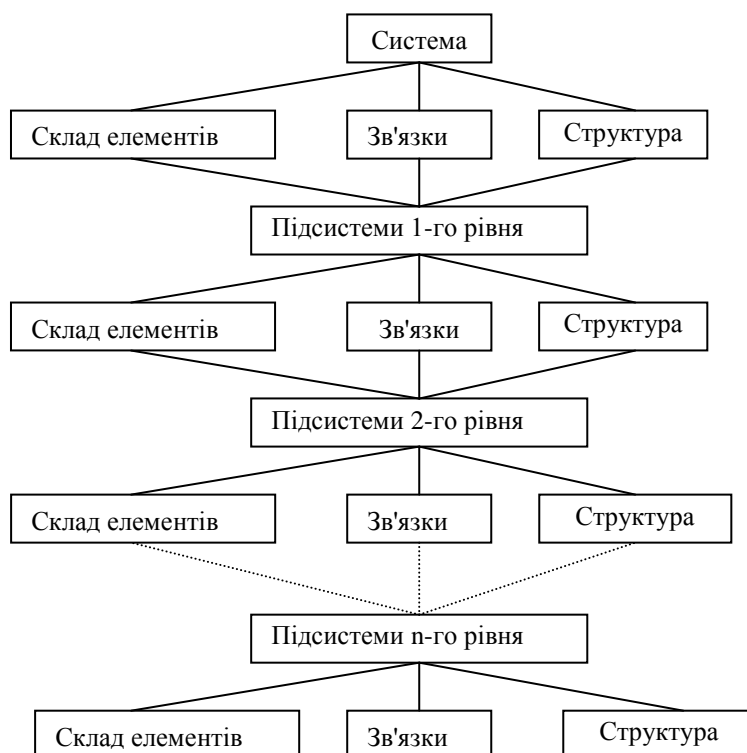


Рис. 1.3. Ієрархія морфологічного опису по вертикалі

Морфологічний опис дає уявлення про будову системи. Глибина опису, рівень деталізації визначається метою і значен-

ням системи. Тут вивчаються і розглядаються елементарний склад, зв'язки та структурні підсистеми на різних рівнях ієрархічного розчленування по вертикалі (рис.1.3).

Функціональний опис (опис функції) так само, як і морфологічний, має ієрархічну схему і відображає ієрархію функцій, процесів і параметрів. Він передбачає вивчення функціональних залежностей між елементами системи, між елементами і системою в цілому, між системою і зовнішнім середовищем.

Інформаційний опис дає уявлення про організацію системи. Він визначає взаємодію морфологічних і функціональних компонент системи залежно від якості і кількості внутрішньої і зовнішньої інформації.

Класифікація систем. Що стосується *класифікації* систем (багаторівневого поділу за якимись принципами), то слід відзначити відсутність у даний час не тільки загальноприйнятого розбиття по групах, але й обов'язково необхідних для цього ознак класифікації. Наприклад, за ступенем взаємодії з навколишнім середовищем (обмін потоками енергії, речовини та інформації) всі системи можуть бути розділені на відкриті, закриті й ізольовані. На відміну від двох останніх, відкриті системи обмінюються зі своїм оточенням всіма цими формами матерії; закриті – лише інформацією, а ізольовані – ні одною з них. Прикладами гомогенних систем можуть служити технічні та організаційні системи, а гетерогенних – людино-машинні (ерготехнічні) системи й етногеотосисистеми (від грец. *ethnos* – народ, *gs* – земля і *ethos* – устрій життя). Відкриті системи можуть бути поділені на рівноважній дисипативні. Останні так названі тому, що вони безперервно розсіюють частину вільної енергії, серед того й у вигляді тепла, що виділяється в навколишнє середовище.

Що стосується ще однієї ознаки – складності систем, то усі вони розділені на три групи: прості, складні і великі. Основними властивостями двох останніх вважаються:

а) унікальність – аналоги помітно відрізняються;

- б) багатоступінчастий склад – ієрархічні підсистеми і компоненти;
- в) випадковий характер функціонування і реагування на вплив різних факторів;
- г) багатокритеріальність оцінки стану – необхідність у векторних показниках якості;
- д) слабка структурованість і різноманітність утворюючих їх частин.

Це означає, що основною відмінною властивістю великої системи є розмірність, що не дозволяє провести її дослідження без попередньої декомпозиції (розчленування на компоненти) з подальшим агрегуванням (укрупненням) їх елементів. Що стосується істотних ознак складної (в сенсі дослідження) і, звичайно ж, великої системи, то до них відносяться багатомодельність, тобто потреба в ансамблі відповідних моделей і методів, та міждисциплінарний характер їх аналізу і синтезу.

Нарешті, завершальною, п'ятою ознакою служить мінливість системи, тобто характер її відгуку по відношенню до впливів різних факторів. У відповідності до цього системи діляться на статичні і динамічні, а також на пасивні (детерміновані) й активні (стохастичні).

Величезна різноманітність видів систем зумовила різноманіття існуючих принципів і підходів до класифікації систем. В основі лежить поняття відношення між суб'єктом (людиною) та об'єктом (навколишнім середовищем). Відповідно до цього підходу всі об'єкти, що існують поза дослідником і відповідають вищезазначеним характеристичним системним властивостям, утворюють об'єктні системи. Об'єктні системи поділяються на два види: природні і штучні (створені людиною).

У системному аналізі суб'єктом є особа, яка приймає рішення (ОПР) – людина, дослідник, колектив однодумців, інститут, держава і т.і., що мають спільні інтереси і єдину мету. По відношенню до об'єктних систем ОПР поділяються на групи: 1) вчені –

природознавці; 2) інженери – винахідники, інженери, проектувальники; 3) керівники, оператори, інженери-технологи і т.і. Загальним для категорій ОПР є те, що свою діяльність вони реалізують у формі прийняття рішень. Процедура прийняття рішення (ППР) – це суб'єктивний процес, який реалізується ОПР. ППР формалізується у вигляді цільової системи (ЦС). ЦС утворює клас суб'єктивних штучних систем, що реалізуються в діалоговому режимі «людина – ЕОМ» у вигляді штучних людино-машинних систем, систем штучного інтелекту, інтелектуальних систем управління.

Системний підхід до вивчення і створення технічних об'єктів припускає рішення трьох основних завдань: **аналіз системи, синтез системи, оцінка і прийняття рішень.**

Аналіз полягає у вивченні властивостей і поведінки систем в різних умовах функціонування, у процесі аналізу встановлюються числові значення показників ефективності систем.

Синтез полягає в побудові можливих варіантів системи. Розрізняють структурний і параметричний синтез. Структурний синтез – це побудова структури системи (елементів і зв'язків). Параметричний синтез – це визначення параметрів елементів при заданій структурі. Завдання синтезу вирішуються при проектуванні технічних систем.

Оцінка і прийняття рішень полягає у виборі найкращого варіанту системи з декількох альтернативних та оцінка ефективності функціонування системи.

Типові технологічні процеси (ТТП) і комплекси з позицій системного аналізу формуються як технологічні системи (Т-системи), які можна охарактеризувати такою групою символів

$$TC = \langle \Phi H, \Phi, CT, K, O, E \rangle,$$

де TC – технологічна система; ΦH – функціональне призначення; Φ – функція; CT – структура; K – компонування; O – організація; E – вектор показників ефективності (якості).

Функціональне призначення (ФП) визначається призначенням проекрованої Т-системи, характеризується об'єктом виробництва, його асортиментом і директивними показниками.

Функція (Ф) Т-системи вводиться для представлення процесів зміни предмета праці, коли останній отримує задані фізичні властивості, і характеризується типовими технологічними процесами. Типовий технологічний процес складається з узгодженої послідовності операцій, що здійснюють фізико-хімічні та інші перетворення продукту згідно з комплексом заданих властивостей.

Функцію Ф зручно описувати орієнтованим графом $G = (TO, U)$, в якому в якості множини вершин $\{TO\}$ виступають технологічні операції, а в якості множини ребер $\{U\}$ – матеріальні зв'язки, що характеризують потоки продукції, що передаються від однієї операції до іншої.

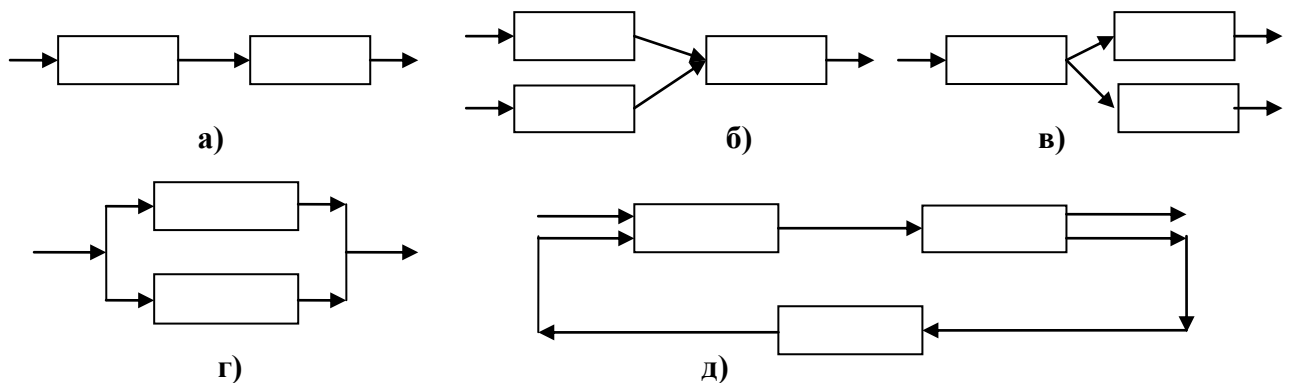


Рис. 1.4. Типові сполучення елементів Т-системи: а) послідовні; б) збіжні; в) розбіжні; г) паралельні; д) із зворотним зв'язком

Під структурою Т-систем (СТ) будемо розуміти деяку організацію за її допомогою синтезу з окремих елементів, що володіють певними властивостями і характеризують мету і призначення системи. СТ відображає якісний і кількісний склад, безліч зв'язків між елементами і визначає основні властивості Т-системи. В якості елемента СТ можна вважати ТО в сукупності з певним

для її кількістю технологічного обладнання (функціональних вузлів) і допоміжними технічними засобами.

На рис.1.4 наведено типові сполучення елементів структури. Структура реальної Т-системи утворюється шляхом взаємодії зазначених сполучень.

Компонування (К) зводиться до задачі геометричного розміщення системних елементів в заданому виробничому об'ємі або на даних виробничих площах. Ефективність компонування оцінюється складністю, коефіцієнтом використання виробничого громадського простору та ін.

Організація Т-системи (О) – це актуалізація і впорядкування зв'язків і самих елементів. Вона полягає в розробці схем взаємодії матеріальних, енергетичних і трудових ресурсів з середовищами виробництва в часі і просторі з метою перетворення предмета праці в заданому напрямку. Найважливішою компонентою є організація управління, покликана забезпечити цілеспрямовану поведінку Т-системи в умовах можливих змін зовнішніх умов і відхилень розрахункових параметрів системи від заданих.

Вектор Е Т-систем визначається сукупністю якісних і кількісних показників ефективності, що характеризує властивості функціонування та структуру системи. Вони можуть бути класифіковані таким чином:

- Показники призначення, які характеризують корисний ефект від використання Т-систем і зумовлює область її застосування;
- Показники надійності і довговічності, що визначають властивість системи зберігати свою працездатність;
- Показники технологічності, що характеризують ефективність конструктивно-технологічних рішень для забезпечення високої продуктивності праці при виготовленні і ремонті Т-систем;
- Показники стандартизації та уніфікації характеризують ступінь використання в системах стандартизованих елементів і рівень уніфікації їх складових частин;

- Економічні показники, що відображають витрати на розробку, виготовлення та експлуатацію Т-систем, а також економічну ефективність їх функціонування.

1.2. Методологія системного моделювання

Найбільш загальною інформаційною моделлю системи є так звана модель "чорного ящика". У цьому випадку система має вигляд прямокутника, внутрішня будова якого прихована від системного аналітика або взагалі невідома. Однак система не є повністю ізольованою від зовнішнього середовища, оскільки останнє має на систему деякий інформаційний або матеріальний вплив. Такі дії отримали назву *вхідних впливів* або *вхідних параметрів* (вхідних змінних). Серед вхідних впливів виділяють спеціальний клас так званих *керуючих впливів*. Останні призначені для того, щоб мати на систему цілеспрямований вплив, призначений для досягнення системою певної мети (цілей) або бажаної поведінки.



Рис. 1.5. Графічне зображення моделі системи у вигляді "чорного ящика"

У свою чергу система також має на середовище або інші системи певні інформаційні або матеріальні впливи, які одержали назву *вихідних впливів* (параметрів, змінних). Графічно така модель може бути зображена на рис. 1.5.

В якості «чорного ящика» принципово можуть бути прийняті різні системи, зокрема нейронні мережі, експертні системи, нечітка логіка, диференціальні рівняння і т.і.

Основне призначення моделі "чорного ящика" полягає в тому, щоб структурувати вихідну інформацію щодо самої системи і зовнішнього по відношенню до неї середовища.

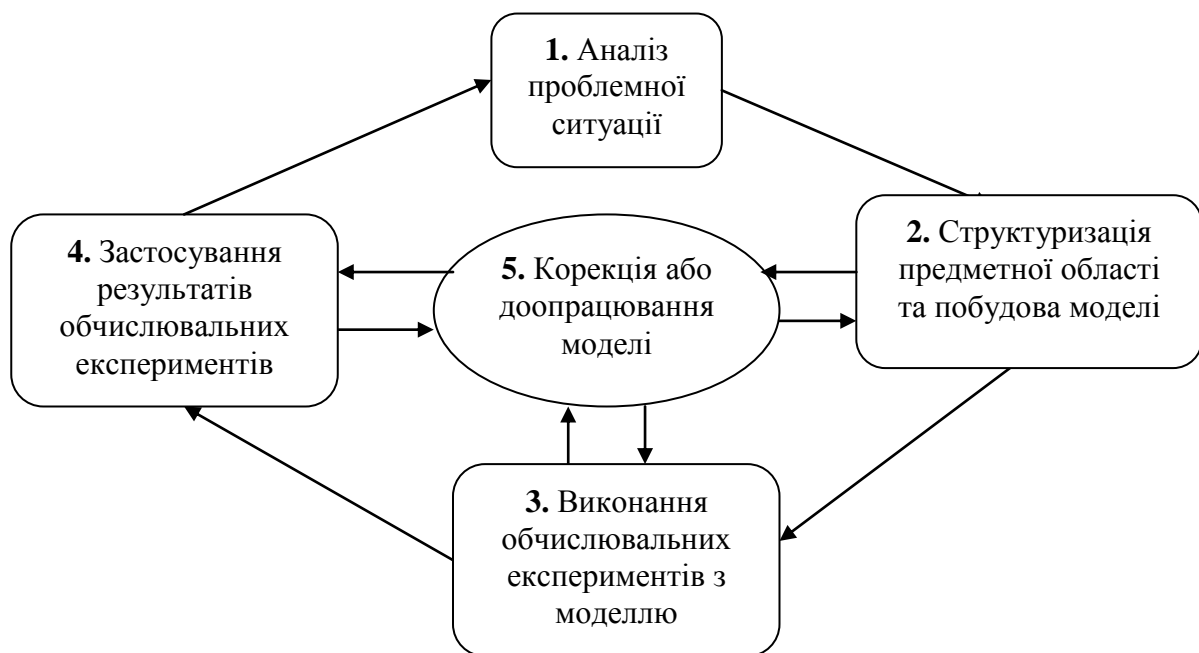


Рис.1.6. Загальна концептуальна схема процесу системного моделювання

У загальному випадку процес системного моделювання може бути представлений у формі взаємопов'язаних етапів, на кожному з яких виконуються певні дії, спрямовані на побудову і подальше використання інформаційно-логічних моделей систем (рис.1.6). Характерною особливістю даного процесу є його циклічний або ітеративний характер, який відображає сучасні вимоги до аналізу і проектування складних систем.

Окремими етапами процесу системного моделювання є:

1. Аналіз проблемної ситуації.

2. Структуризація предметної області та побудова моделі.
3. Виконання обчислювальних експериментів з моделлю.
4. Застосування результатів обчислювальних експериментів.
5. Корекція або доопрацювання моделі.

Головне призначення першого етапу (аналіз проблемної ситуації) – логічне осмислення розв'язуваної проблеми в контексті методології системного моделювання. Тут виконується аналіз всіх наявних ресурсів (матеріальних, фінансових, інформаційних та інших), необхідних для побудови моделі, її використання та реалізації отриманих результатів з метою вирішення наявної проблеми.

У разі відсутності необхідних ресурсів на даному етапі може бути прийняте рішення або про звуження розв'язуваної проблеми, або взагалі про відмову від використання засобів системного моделювання.

Початковий аналіз розв'язуваної проблеми та відповідної проблемної області є найменш формалізованим з точки зору застосування відомих аналітичних підходів і засобів. Тому на даному етапі часто застосовують евристичні або неформальні методи системного аналізу, до яких відносяться:

- методи побудови логічних сценаріїв природною мовою для аналізу можливих способів і альтернативних шляхів вирішення проблеми;

- методи побудови й аналізу дерева цілей і завдань, які дозволяють розбити вихідну проблему на ряд більш приватних або більш простих підпроблем;

- методи морфологічного та концептуального аналізу для досягнення необхідної повноти розгляду вихідної проблеми.

Метою структуризації предметної області та побудови моделі є побудова адекватної моделі системи у відповідній предметній області. Інформація надається у формі моделі системи або проблемної області в цілому певною формально-логічною мовою

(інформаційно-логічна модель системи). Даний етап передбачає виконання такої послідовності дій:

1. Побудова концептуальної або інформаційної моделі системи і проблемної області, яка містить найбільш загальну інформацію і відображає структурні взаємозв'язки системи-оригіналу з іншими об'єктами навколишнього середовища.

2. Побудова аналітичної або математичної моделі системи, яка деталізує окремі аспекти структури і поведінки системи-оригіналу у формі тексту з використанням математичної символіки.

3. Побудова імітаційної або програмної моделі системи, яка реалізує інформаційно-логічну модель.

Для побудови моделей використовуються формально-теоретичні методи, засновані на подальшому розвитку математичних і логічних засобів моделювання.

Базовим об'єктом для отримання нової інформації про систему-оригінал є програмна модель складної системи, реалізована однією з мов програмування або побудована з використанням відповідних програмних інструментаріїв.

Реалізація даного етапу в контексті методології системного моделювання означає виконання серії експериментів з програмною моделлю:

1. Формування конкретних значень наборів вихідних даних, що характеризують окремий обчислювальний експеримент.

2. Виконання розрахунків або виконання окремої ітерації з імітаційною моделлю системи з метою отримання конкретних значень вихідних параметрів моделі.

3. Оцінка точності і верифікація (істинність) отриманих результатів з використанням аналітичної моделі.

4. Інтерпретація отриманих результатів у вигляді керуючих впливів.

5. Оцінка потенційної можливості реалізації отриманих результатів стосовно системи-оригіналу.

Корекція або доопрацювання моделі вимагає внесення змін в існуючу модель, спрямовані на забезпечення адекватності розв'язуваної проблеми. При цьому складність системи і, відповідно, її моделі можуть бути розглянуті з різних точок зору. Насамперед, можна виділити складність структури системи, яка характеризується великою кількістю елементів системи і різними типами взаємозв'язків між цими елементами.

Системний аналіз являє собою широку стратегію наукового пошуку з використанням математичного апарату і математичних моделей. Сутність системного аналізу визначається його стратегією, в основі якої лежать загальні принципи, застосовні до вирішення будь-якої системної завдання. З позицій системного аналізу вирішуються задачі моделювання, оптимізації, управління та оптимального проектування технологічних процесів і технологічних систем у масштабі виробництва. Сутність системного підходу в даному випадку полягає в тому, що вся апріорна інформація, а також отримана експериментально в лабораторних умовах, на дослідних і промислових установках, послідовно накопичується і збагачується в процесі розробки математичної моделі технологічної системи. Побудована математична модель після встановлення адекватності її досліджуваному об'єкту використовується для аналізу, синтезу, оптимізації та керування системою.

Сучасне виробництво (хімічне, металургійне, харчове тощо) представляє собою ієрархічну структуру по горизонталі: підготовка сировини, хімічне перетворення і виділення продуктів. Кожна зі стадій може містити довільну кількість різнорідних процесів, що відрізняються природою визначальних явищ, а саме: а) гідродинамічні процеси: переміщення рідин і газів в апаратах і трубопроводах, отримання і розділення неоднорідних систем газ – рідина (тумани), газ – тверде речовина (пилу), рідина – тверда речовина (суспензії), рідина - рідина (емульсії); б) теплові процеси: кипіння, випаровування, конденсація випарювання; в) дифузійні процеси: екстракцію, абсорбцію, кристалізацію, мембранні, рек-

тифікацію тощо; г) хімічні процеси: хімічні перетворення в реакторах; д) біохімічні процеси: біохімічні перетворення в реакторах, аеротенах і т.і.; е) механічні процеси: подрібнення твердих речовин та їх транспортування, сепарування сипучих речовин і т.і.

1.3. Формалізація системи. Структура детерміновано-стохастичної моделі

Аналіз технологічного процесу включає в себе: 1) математично-логічне формулювання задачі для заданої фізичної ситуації; 2) детальний аналіз з метою побудови математичних моделей; 3) синтез та опис результатів для досягнення повного розуміння процесу. Під процесом будемо розуміти серію реальних операцій або обробок вихідних матеріалів, а під моделлю – математичний опис цього реального процесу, який допомагає аналізувати їх і робити обґрунтовані прогнози.

Формалізація системи здійснюється за допомогою математичної моделі, що виражає зв'язок між вихідними і вхідними параметрами системи, керуючими і збуреними змінними. Складна система зазвичай формалізується як детерміновано-стохастична модель [3,4], структура якої показана на рис. 1.7.

Детермінованими моделями називаються моделі, в яких кожна змінна або параметр може приймати певне фіксоване значення або ряд фіксованих значень у будь-яких заданих умовах, або, іншими словами, такі системи, у яких існує однозначний зв'язок між вхідними і вихідними змінними. У статистичних (стохастичних), або імовірнісних моделях допускається невизначеність. Змінні параметри, використовувані для опису зв'язків між входом і виходом, а також структура елементів (обмежень) точно не відомі, тобто зміна визначальних величин відбувається випадково і при цьому значення вихідної змінної не є однозначним відповідно до вхідної.

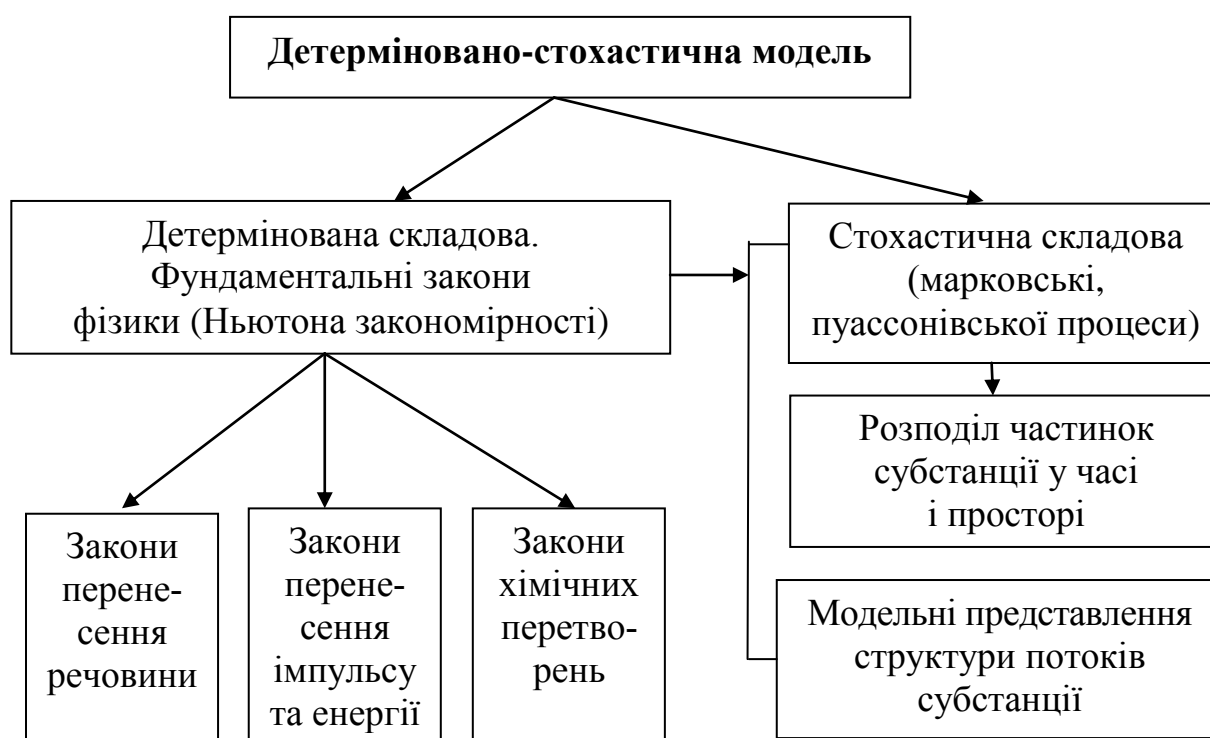


Рис. 1.7. Структура детерміновано-стохастичної моделі

Різні моделі технологічних процесів можна розбити на три групи:

- 1) моделі явищ переносу, засновані на фізико-хімічних принципах;
- 2) моделі балансу популяцій, що базуються на балансі популяцій;
- 3) емпіричні моделі, що застосовуються для підгонки експериментальних даних.

Розподіл за часом перебування та інші розподіли за віком є прикладами моделей балансу популяцій. Типовими прикладами емпіричних моделей служать поліноми, які використовуються при підгонці експериментальних даних.

В табл. 1.1 моделі явищ переносу класифіковані за ступенем складності фізичних уявлень про процес, необхідних для побудови моделі; ця складність зменшується зверху вниз.

Таблиця 1.1. Моделі явищ переносу за складністю фізичних уявлень

Рівень фізико-хімічних уявлень	Поширеність серед дослідників	Основні застосування	Аналізовані параметри
Молекулярно-атомний	Використовується при вивченні механізму явищ	Розгляд дискретних величин, квантова механіка, статистична механіка, кінетична теорія	Функції розподілу, інтеграли зіткнень
Мікроскопічний	Застосовується тільки в особливих випадках	Явища ламінарного перенесення, статистичні теорії турбулентності	Кінетичні коефіцієнти; коефіцієнти в'язкості, дифузії, теплопровідності
Ураховуються просторові неоднорідності	Застосовується тільки в особливих випадках	Явища ламінарного і турбулентного переносу, перенесення у пористих середовищах	"Ефективні" коефіцієнти переносу
Ураховуються обмеження на максимальний градієнт	Використовується для проточних систем, поршневих потоків	Явища ламінарного і турбулентного перенесення, розрахунки реакторів	Міжфазні коефіцієнти перенесення, кінетичні константи
Макроскопічний	Широко використовується	Технологічні процеси, загальні процеси, класична кінетика і термодинаміка	Міжфазні коефіцієнти переносу, макроскопічні кінетичні константи, коефіцієнти тертя

У табл. 1.2 наведено класифікацію явищ переносу [5], засновану на тип рівнянь, що використовуються в моделях; таким чином, вона поділяє моделі за ступенем складності їх вирішення. Установлений стан означає рівність нулю похідних за часом. У разі зосереджених параметрів передбачається, що можна знехтувати їх зміною в просторі; різні властивості і стан системи (залежні змінні) можуть вважатися однаковими по всій системі. Наявність розподілених параметрів, навпаки, передбачає детальний облік зміни поведінки при переході від однієї точки до іншої.

Таблиця 1.2. Детерміновані моделі переносу, засновані на їх математичній структурі



Всі реальні системи, звичайно, є системами з розподіленими параметрами в тому сенсі, що в них завжди присутні які-небудь неоднорідності. Однак ці неоднорідності нерідко виявляються відносно малопомітними, так що ними можна знехтувати і тоді система буде мати зосереджені параметри.

Якщо змінна величина Y на виході підсистеми повністю визначається величиною X на вході (до вхідних змінних можна віднести склади і температури фаз, що надходять на фізико-хімічну переробку, швидкість, густина і т. і.) , параметрами підсистеми і початковими і граничними умовами, то в межах фізико-хімічної

системи (ФХС) вхідні змінні X зазнають цілеспрямоване фізико-хімічне перетворення у змінні Y (рис. 1.8).

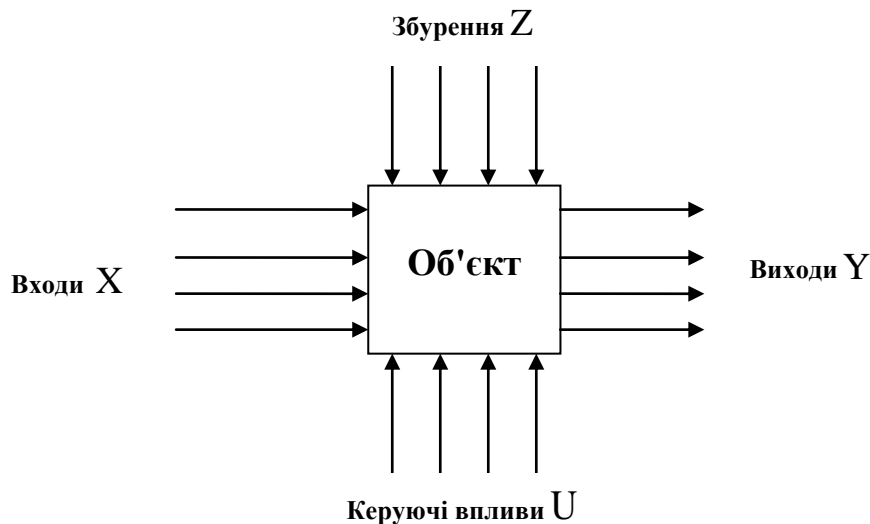


Рис.1.8. Найпростіша структура системи

З математичної точки зору цьому перетворенню відповідає його математична модель

$$Y = \Phi(X) \quad (1.1)$$

у формі функціонального оператора Φ , який відображає функціональний простір вхідних змінних X у простір значень вихідних змінних Y .

Модель (1.1) є ідеалізацією моделі, що відображає ступінь знань про процес

$$Y = T(X), \quad (1.2)$$

тобто вектор X (температура, швидкість, густина і т. і.) під дією деякого технологічного оператора T перетворюється на вектор вихідних змінних.

Складність структури оператора T полягає в тому, що він є накладенням цілого ряду "елементарних" технологічних операторів: конвективного і турбулентного переносу тепла і речовини; дифузійного, хімічного перетворення; дроблення, подрібнення, кристалізації, змішування і т.і. У загальному випадку технологіч-

ний оператор T відображає сукупність лінійних, нелінійних розподілених у просторі та змінних у часі процесів і має змішану детерміновано-стохастичну природу.

Вектор дійсних вихідних змінних (1.2) не збігається з виходом моделі (1.1) через те, що оператор Φ є наближеною характеристикою оператора T .

У явній формі оператор, що здійснює відображення (1.1), являє собою замкнуту систему диференціальних, інтегральних, інтегро-диференціальних рівнянь і співвідношень емпіричного характеру, доповнену необхідними початковими і граничними умовами.

Для ефективного вирішення завдань, що виникають на всіх рівнях ієрархії виробництва, необхідно, насамперед, ідентифікувати оператори окремих ФХС, що складають технологічну систему (ТС), тобто оцінити вхідні в них параметри. Це може бути досягнуто або рішенням обернених задач з постановкою відповідних експериментів, або апріорним вибором орієнтовних значень технологічних параметрів з використанням даних аналогічних виробництв (при проектуванні нових технологічних систем). Після процедури ідентифікації можна вважати, що відображення (1.1) готове для використання при вивченні властивостей ФХСу робочому діапазоні зміни її параметрів, знаходження оптимальних конструктивних і режимних параметрів технологічного процесу; синтезу оптимального керування системою, аналізу і моделювання поведінки ХТС. Реалізація наведених завдань так чи інакше пов'язана з розв'язанням системи рівнянь, відповідних відображенню (1.1), що рівнозначно отриманню явного функціонального зв'язку між змінними Y і X або в аналітичній формі кінцевих співвідношень, або у вигляді результату чисельного рішення задачі на ЕОМ. Формально цей розв'язок запишемо у вигляді

$$Y = \varphi(X), \quad (1.3)$$

в якому вектор-функція j відображає результат аналітичного або чисельного розв'язання наведеної системи рівнянь.

Відображення (1.3) несе в собі важливу інформацію про спосіб (метод) одержання цих співвідношень: це конкретний метод аналітичного розв'язку системи (1.1), або алгоритм чисельного розв'язання задачі з повним комплектом програмного забезпечення для його реалізації на ЕОМ.

Визначене таким способом відображення (1.3) називається модулем ФХС або модульним оператором ФХС:

$$\varphi = \{\Phi, M\},$$

де M – метод розв'язання функціонального оператора.

Практика системних досліджень показує, що для ефективного вирішення завдань вищих рівнів ієрархії виробництва (аналіз і синтез ХТС, оптимізація та управління ХТС, автоматизоване проектування), найкращий модульний принцип подання інформації, що надходить з нижніх рівнів ієрархії виробництва.

Стратегія системного підходу до дослідження технологічного процесу на першому етапі передбачає якісний аналіз структури ФХС (рис. 1.9), з якого виділимо два аспекти: змістовий, тобто попередній аналіз апріорної інформації про фізико-хімічні особливості процесу, і математичний, тобто якісний аналіз структури математичних залежностей, які можуть бути покладені в основу опису ФХС [3].

Виділення підсистем (рівнів) визначається не тільки складністю досліджуваного об'єкта, але й ступенем вивченості даного рівня і наявністю математичного опису. Розглядаючи незалежно кожен з підсистем з вхідними та вихідними потоками (енергії, імпульсу, маси і т.і.) й оцінюючи потенціал цих потоків, можна виявити джерела і стоки, визначити допустимі за деяким критерієм втрати, а також виявити резерви підвищення ефективності окремих апаратів і схеми в цілому.

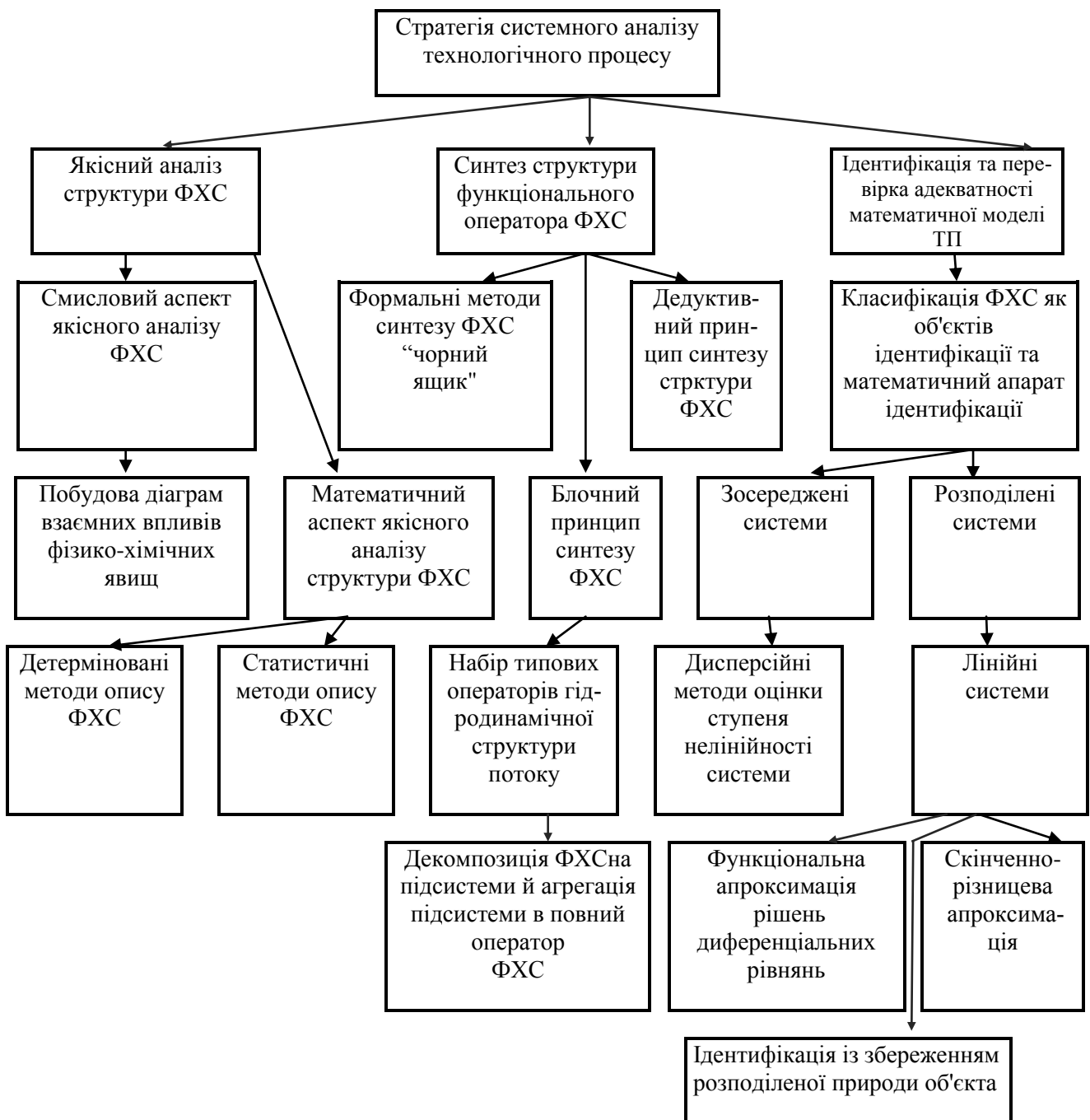


Рис. 1.9. Блок-схема стратегії системного аналізу ТП

1.4. Загальна схема процесу математичного моделювання. Блочний принцип побудови математичної моделі

Для однієї і тієї ж системи існують різні математичні моделі, кожна з яких служить для вирішення приватної задачі, пов'язаної з вивченням цієї системи. Два великих класи утворюють статичні моделі (похідні за часом дорівнюють нулю) і динамічні моделі (поведінка системи під час перехідного процесу), і в кожному з них ступінь деталізації залежить як від розв'язуваної задачі, так і від кількості наявних вихідних даних. Точний опис хіміко-технологічних процесів (найбільш повна модель) часто призводить до великої системи громіздких рівнянь. Хоча вони і можуть бути вирішені чисельними методами, але для аналітичного дослідження бажано, на основі загальних міркувань, звести цю систему до більш простої. Рішення такої системи для всіх практичних випадків буде цілком задовільним і точність його визначається вихідними даними.

Одним з найважливіших аспектів математичного моделювання є процес складання рівнянь. Практика обчислень показує, що якщо вся сукупність рівнянь знаходиться в логічному або причинно-наслідковому зв'язку, то обчислювальна модель є стійкою. Ця логічна послідовність рівнянь, що відповідає природі та сутності спостережуваного явища, називається природним розташуванням. Ключем до розуміння внутрішнього механізму процесів є здатність дослідника виявити в них цей природний причинно-наслідковий зв'язок.

До появи потужних обчислювальних машин сучасних поколінь, поширеними методами аналізу було зведення задачі до вирішення елементарних систем рівнянь. При цьому надмірно спрощувалася розв'язувана задача. Багато часу витрачалось на оволодіння математичними прийомами і методами розв'язування диференціальних рівнянь. Практика свідчить, що лише незначна частина інженерів у своїй практичній діяльності використовує

математичні методи вирішення складних систем рівнянь (аналітичні методи).

Таблиця 1.3. Класифікація математичних проблем та можливість їх вирішення аналітичними методами

Вид рівняння	Типи рівнянь					
	Лінійні			Нелінійні		
	Одне	Кілька	Багато	Одне	Кілька	Багато
Алгебраїчне	Просто	Легко	Важко або неможливо	Дуже важко	Дуже важко	Неможливо
Звичайне диференціальне рівняння	Легко	Важко	Важко або неможливо	Дуже важко	Неможливо	Неможливо
Рівняння у приватних похідних	Важко	Важко або неможливо	Неможливо	Неможливо	Неможливо	Неможливо

Тут доречно навести частково класифікацію (табл.1.3) математичних рівнянь [6], що мають аналітичне рішення. Рівняння, які можна вирішити аналітично, через свою тривіальність мають мале практичне застосування. Більшість проблем, пов'язаних з дослідженням сучасних промислових процесів, призводить до складних систем нелінійних диференціальних рівнянь великої розмірності, які, як правило, аналітичного рішення не мають і можуть бути вирішені лише за допомогою ЕОМ. Математична модель являє собою деяку сукупність математичних співвідношень, які відображають поведінку об'єкта моделювання при наявності обмежень на параметри процесу та з урахуванням впливу на об'єкт зовнішніх факторів.

Процес математичного моделювання і дослідження системи можна умовно розбити на сім стадій (рис.1.10).

Перша стадія, найбільш важлива, – це постановка завдання, оскільки не існує загальних правил, які були б досить корисні в усіх випадках. Технічні проблеми настільки різноманітні, що для

успіху аналізу повинна бути зрозуміла природа даної конкретної задачі. Постановка задачі визначає не лише мету аналізу, але і шляхи вирішення завдання.

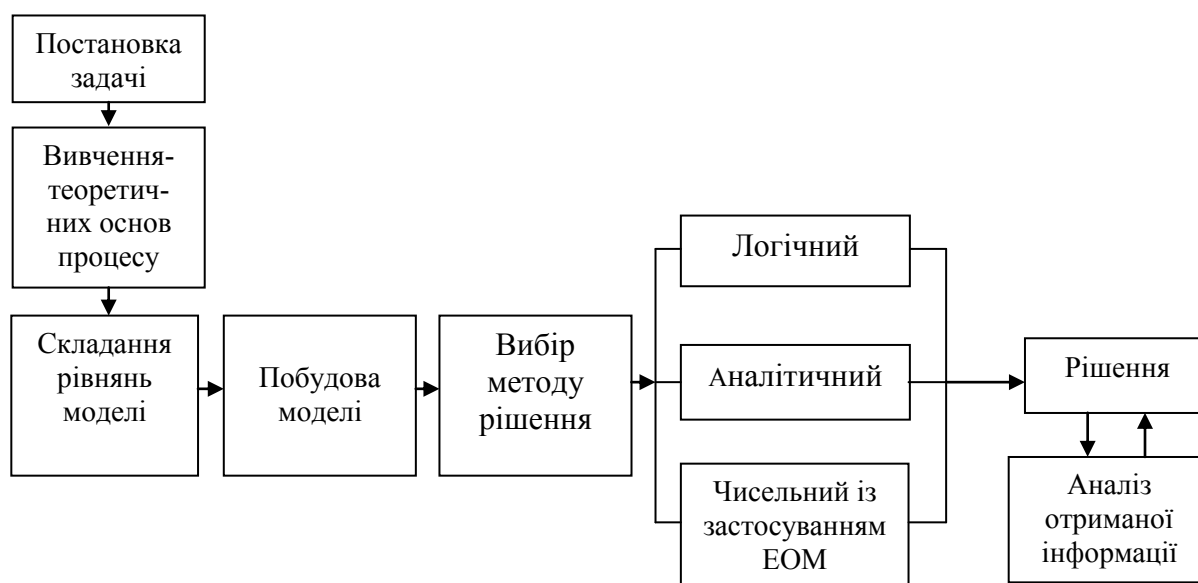


Рис. 1.10. Загальна схема математичного моделювання

Друга стадія – це визначення фундаментальних законів, яким підпорядковується механізм явищ, що лежать в основі проблеми. Теоретичні основи процесів вивчаються зазвичай з різних джерел, опублікованих і неопублікованих. Якщо не вдається підібрати задовільну теорію, то вдаються до постулатів. Справедливість останніх перевіряється порівнянням результатів розв'язків рівнянь математичної моделі, побудованої на основі прийнятих постулатів, з експериментальними даними. Таким же чином з'ясовується, яка з кількох можливих теорій правильніше відображає сутність досліджуваних явищ.

Третя стадія. На основі вибраної фізичної моделі стосовно до розв'язуваної задачі записується система відповідних математичних рівнянь. На даному етапі не потрібно виконувати ніяких дій, окрім, якщо це можливо, знехтування незначними членами. Відкидаючи їх, треба бути обережним і переконатися в тому, що

не включений член дійсно незначний протягом усього ходу рішення задачі. Добре відомо, що необґрунтоване нехтування різного роду "малими величинами" при математичному моделюванні реальних процесів може істотно спотворити справжню картину явищ. Тому в системах рівнянь, що описують ще не вивчений процес, дослідник змушений враховувати велику кількість, на перший погляд, другорядних факторів. Наслідком цього, як правило, є з одного боку – відносно високий порядок системи, з іншого – її жорсткість (явище жорсткості виявляється і визначається необхідністю одночасного залучення для адекватного опису процесів у будь-якій точці відрізка спостереження швидкоплинних робочих функцій з великими значеннями похідних і функцій з малими значеннями похідних). При подальшому вивченні процесу ступінь жорсткості і порядок системи зазвичай знижується, але існує і ряд важливих завдань, де жорсткість присутня по самій суті речей. Іноді можна виключити з розгляду цілі рівняння, нехтуючи малими відхиленнями деяких змінних. Наприклад, припустимо, що питома теплоємність багатоконпонентної суміші, що входить в рівняння теплового балансу, змінюється, через малі зміни складу суміші, всього лише на 1 % від номінального значення. Таким чином, перш ніж включити рівняння в математичну модель, треба оцінити вплив вхідних у нього змінних на результат моделювання і, за можливості, замінити змінні, які мало впливають на процес постійними середніми величинами.

Четверта стадія. Побудова моделі – найбільш тонка і відповідальна частина математичного моделювання і є основою всього системного аналізу. Це центральний етап дослідження або проектування будь-якої системи. При цьому потрібно не тільки і не стільки знання математики, скільки глибоке розуміння суті описуваних явищ. Від якості моделі залежить доля всього подальшого аналізу. Побудова моделей завжди процедура неформальна і, звичайно, дуже сильно залежить від дослідника, його досвіду, таланту. Вона завжди спирається на певний дослідний мате-

ріал, у зв'язку з чим ми говоримо, що процес моделювання має феноменологічну основу. Модель має досить правильно відображати явища, однак одного цього ще мало. Вона повинна бути зручною для використання. Тому ступінь деталізації моделі, форма її подання і т.і. визначаються цілями дослідження і безпосередньо залежать від дослідника. Працюючи з одним і тим же дослідним матеріалом, різні дослідники можуть представляти його зовсім по-різному. Але при всьому тому відмінності існують загальні принципи моделювання, ігнорування яких неприпустимо.

Основним прийомом побудови математичної моделі досліджуваного об'єкта служить блоковий принцип. Згідно з цим принципом, після того як набір елементарних процесів встановлений, кожний з них досліджується окремо (по блоках) в умовах, максимально наближених до умов експлуатації об'єкта моделювання. В результаті кожному елементарному технологічному оператор ставиться у відповідність елементарний функціональний оператор з параметрами, досить близькими до істинних значень.

При моделюванні технологічних процесів спочатку досліджується гідродинамічна частина загального технологічного оператора – основа майбутньої моделі. Ця частина оператора відображає поведінку так званого "холодного" об'єкта, тобто об'єкта без фізико-хімічних перетворень, але з реальними навантаженнями на апарат по фазах. Важливо підкреслити, що відповідний елементарний функціональний оператор тут, як правило, лінійний і являє собою або лінійні диференціальні рівняння, або лінійні інтегральні перетворення з ядром у вигляді функції розподілу елементів потоку за часом перебування в апараті.

Далі вивчають кінетику хімічних реакцій, швидкість процесів масо- і теплопередачі, кінетику фазових переходів в умовах, близьких до умов експлуатації об'єкта, і складають відповідні елементарні функціональні оператори. Ці елементарні процеси зазвичай є основними джерелами нелінійностей результуючого

функціонального оператора (хімічні реакції порядку, відмінного від нуля та одиниці, нелінійні рівноважні співвідношення, експоненціальна залежність кінетичних констант від температури і т.і.).

Наступне завдання полягає в агрегуванні елементарних функціональних операторів у загальний результуючий функціональний оператор, який і представляє математичну модель об'єкта. Важливим фактором агрегування є правильна координація окремих операторів між собою, яка не завжди можлива через труднощі урахування природних причинно-наслідкових зв'язків між окремими елементарними процесами.

Коли рівняння складені, визначається метод вирішення їх спільної системи. Спочатку треба провести природне розташування рівнянь за допомогою побудови блочної поточно-інформаційної діаграми (схема зв'язку окремих стадій технологічного процесу). Діаграма будується так, щоб було чітко видно, як може бути використане кожне рівняння, з якою метою перебуває кожна змінна і які внутрішні зв'язки між ними.

Є два важливі міркування на користь природної моделі порівняно з іншими можливими математичними побудовами. Перше з них полягає в тому, що природна модель, заснована безпосередньо на причинно-наслідкових зв'язках, допоможе досліднику у з'ясуванні механізму досліджуваного фізичного явища. Друге полягає в тому, що неприродна з цієї точки зору модель веде до розбіжності розв'язків, у той час як природній моделі властива, як правило, обчислювальна стійкість.

Цей метод є простим поширенням на системи нелінійних рівнянь класичної схеми зображення спільних систем лінійних диференціальних рівнянь з допомогою передавальних функцій. Така схема, як і логічний причинно-наслідковий зв'язок у фізичній системі, дає ясну картину передбачуваного механізму досліджуваного явища, а іноді виявляє зв'язки між змінними, які не були очевидними на попередній стадії.

Взагалі кажучи, існує кілька підходів до побудови моделі, але можна сформулювати такі основні положення, які можуть бути корисні при будь-якому підході:

1. Записують рівняння, що зв'язують істотні для процесу величини.

2. Відповідно до природних причинно-наслідкових зв'язків досліджуваного процесу вирішують, яким чином має бути використане кожне рівняння (тобто, яку невідому величину треба визначити з кожного рівняння).

3. Вводять граничні умови і початкові значення для всіх змінних, що входять у рівняння системи. При цьому повинно бути стільки початкових значень, скільки є диференціальних рівнянь (у перерахунку на диференціальні рівняння першого порядку).

4. Розташовують рівняння в будь-якій послідовності, маючи на увазі, однак, що повинні зберігатися основні зв'язки, що існують між потоками інформації в досліджуваній системі.

5. Проводять остаточну перевірку для того, щоб переконатися, що всі величини, необхідні для вирішення кожного окремого рівняння щодо виділеної змінної, визначаються з якого-небудь іншого рівняння моделі або є відомими константами.

Обчислювальній стадії передують ще один етап – аналіз інформації, яку ми хочемо отримати при вирішенні рівнянь моделі. Складаючи таблицю різних випадків, розглянутих для даної задачі, та інформації, яка очікується в кожному випадку, можна виявити надлишкові ситуації і полегшити складання програми розрахунку на наступній стадії.

Після складання математичного опису та вибору відповідних початкових та граничних умов необхідно вибрати метод рішення і скласти програму (алгоритм).

Цей етап хоча і трудомісткий, але у більшою мірою, ніж будь-який інший, піддається формалізації на ЕОМ з розвинутим

математичним забезпеченням і для низки типових задач може виконуватися автоматично.

У простих випадках, коли можливе аналітичне рішення системи рівнянь математичного опису, необхідність спеціальної розробки моделюючого алгоритму відпадає, так як вся інформація виходить з відповідних аналітичних рішень. Коли математичний опис являє собою складну систему кінцевих диференціальних рівнянь, від можливості будови досить ефективного моделюючого алгоритму може істотно залежати практична придатність математичної моделі. Особливо це важливо при використанні моделі для вирішення задач, в які вона входить в якості складової частини більш загального алгоритму, наприклад, алгоритму оптимізації. Як правило, в таких випадках для реалізації математичної моделі доводиться застосовувати засоби обчислювальної техніки, без яких фактично не можна ставити і вирішувати скільки-небудь складні завдання математичного моделювання і, тим більше, задачі оптимізації, де розрахунки за рівняннями математичного опису зазвичай повторюються багато разів.

П'ята стадія. Вибирається один з кількох можливих способів рішення залежно від рівня проведеного дослідження процесу і від складності рівнянь моделі. Слід зазначити, що дослідження об'єктів, що описуються диференціальними, інтегральними та інтегро-диференціальними рівняннями, методом математичного моделювання являє іноді досить складну обчислювальну задачу. Тому в ряді випадків замість математичного опису об'єкта диференціальними або інтегральними рівняннями його характеризують системою кінцевих рівнянь, для чого від безперервного об'єкта з розподіленими параметрами переходять до дискретного з зосередженими параметрами, але з так званою комірковою структурою. Формально заміна безперервного об'єкта дискретним еквівалентна заміні диференціальних рівнянь різницевиими співвідношеннями, а інтегральних – алгебраїчними рівняннями. При цьому для об'єктів, що описуються звичайними диференціа-

льними рівняннями, математичний опис представляють у вигляді системи кінцево-різницевих рівнянь. Для процесів, що характеризуються диференційними рівняннями в приватних похідних, результатом є система диференціально-різницевих рівнянь. При таких перетвореннях вихідної системи рівнянь, природно, допускається похибка, яку необхідно враховувати при оцінці результатів моделювання.

Разом з тим існує ряд об'єктів, які за своєю природою мають комірчасту структуру. Тому комірчасті моделі відіграють роль не тільки кінцево-різницевої апроксимації диференціальних операторів об'єктів, але і мають цілком визначене самостійне значення, відображаючи комірчасту структуру реального об'єкта.

При виборі методу розв'язання рівнянь математичного опису зазвичай ставиться задача забезпечення максимальної швидкодії (особливо при вирішенні завдань, пов'язаних з управлінням об'єкта) при мінімумі займаної програмою пам'яті ЕОМ. Природно, що при цьому повинна забезпечуватися задана точність рішення.

Більшість методів є ітераційними. Це означає, що рішення виходить як межа деякої послідовності наближень. Якщо ця послідовність наближень прагне до межі, то обчислювальний процес називається збіжним, а якщо ні – то розбіжним.

Використання ітераційних методів відповідає вимозі міні-оптимізації займаної пам'яті ЕОМ, так як послідовні наближення виконуються за одними і тими ж формулами, але з різними цифровими даними. Аналогічний порядок обчислень можливий і при застосуванні рекурентних співвідношень.

Перш ніж вибрати той чи інший чисельний метод, необхідно проаналізувати обмеження, пов'язані з його використанням, наприклад, піддати функцію або систему рівнянь аналітичного дослідження, у результаті якого з'являється можливість використання даного методу. При цьому досить часто вихідна функція чи система рівнянь повинна бути відповідним чином перетворена, з тим, щоб можна було ефективно застосувати чисельний метод.

Перетворенням або введенням нових функціональних залежностей часто можливо значно спростити завдання.

Істотним моментом при виборі методу є розмірність задачі. Деякі методи ефективні при вирішенні невеликих завдань, однак, із збільшенням числа змінних обсяг обчислень настільки зростає, що доводиться від них відмовлятися. Такого класу завдання зазвичай мають місце при вирішенні систем рівнянь, пошуку оптимальних значень параметрів багатовимірних функцій. Відповідним вибором методу можна зменшити час розв'язання задачі і об'єм займаної пам'яті машини.

Шоста стадія – аналіз моделі. Фактично можна виділити три основних рівні аналізу моделі. Якщо треба вирішити нескладний запит і рівняння досить прості, то відповідь отримують шляхом перегляду моделі, не вирішуючи рівняння, що входять до неї. Ясно, що цей метод не може бути поширений на більш складні випадки без значного збільшення кількості невипробуваних припущень.

Наступний рівень аналізу знову дуже обмежений завданнями помірної складності, пов'язаний з рішенням рівнянь аналітичними методами. Щоб вирішити навіть найпростіші системи нелінійних рівнянь, потрібна висока кваліфікація.

Третій рівень аналізу, що проводиться з використанням ЕОМ, являє собою найбільш результативний, єдино доцільний для завдань високої складності шлях.

Сьома стадія – вивчення і підтвердження результатів, отриманих при вирішенні рівнянь моделі. Будь-якому, не передбачуваному заздалегідь рішенню має бути дано пояснення, щоб гарантувати себе від помилок, які можуть виявитися в результаті обчислень. Цикли обчислень необхідно будувати так, щоб можна було перевірити правильність ходу рішення рівнянь моделі.

Математична модель об'єкта є лише його визначенням у рамках прийнятих допущень аналогом. Тому значення змінних, отриманих на моделі і об'єкті, розрізняються. Етап встановлення

адекватності (відповідності) моделі об'єкта є заключним в послідовності етапів, виконуваних при її розробці. Результати, отримані на цьому етапі, можуть призвести до істотних змін завдання, починаючи з постановки.

Математична модель після встановлення адекватності надходить в експлуатацію і як прикладна програма може бути використана у двох варіантах: автономно або в сукупності з іншими програмами. Проте, незалежно від способу застосування прикладна програма зазвичай оформляється відповідно до певних вимог так, щоб забезпечити можливість її зберігання і автономного або системного використання. В подальшому виконання цієї програми буде проводитися під управлінням спеціальної програми. Відмінність у використанні буде проявлятися в тому, що в автономному режимі вона використовується як одиночна програма, а в системному – як елемент системи взаємопов'язаних програм.

Ідея побудови математичних моделей шляхом поєднання більш простих елементів знайшла втілення у блоково-модульному принципі моделювання. Під модулем мається на увазі деяка частина завдання, яка може бути проаналізована окремо, а під блоком модулів – таке їх поєднання, яке служить єдиній меті. При моделюванні технологічних процесів будемо виділяти такі блоки: блок математичної моделі агрегату; блок зв'язків і апаратів, складових агрегату; блок критеріїв оптимізації; блок алгоритмів оптимізації та ін. Всі блоки являють собою ієрархічні структури; на кожному рівні ієрархії виділяють кілька варіантів модулів. Вся система є відкритою, завдяки чому можна, за необхідності, вводити нові блоки, блоки - нові рівні, на кожному рівні – нові модулі і т.і. З іншого боку, блок повинен бути гнучким, щоб можна було деякі модулі включати або відключати залежно від типу розв'язуваних задач.

Програма, призначена для автономного та системного використання й оформлена відповідно до певних вимог, також називається модулем. Правила оформлення модулів залежать від

особливостей системи, в якій вони будуть використовуватися, а також від мови, якою вони написані. Подання прикладних програм у вигляді модулів по суті є формою уніфікації правил їх складання. Це полегшує правила їх використання в різних за призначенням системах, спрощує їх об'єднання з іншими модулями. Для зазначення характеристик кожен модуль повинен супроводжуватися свого роду паспортом, в якому може міститися така інформація:

- 1) опис задачі, математичне формулювання з переліком прийнятих допущень і опис алгоритму рішення;
- 2) назва модуля і назва мови, якою він написаний;
- 3) перелік і призначення вхідних і вихідних параметрів;
- 4) перелік і призначення внутрішніх або проміжних параметрів;
- 5) опис схем для реалізації багатоцільових модулів із зазначенням входу і виходу для кожної схеми;
- 6) зазначення вводу-виводу з визначенням уведених і виведених змінних;
- 7) зазначення характеристик за швидкістю, обсягом займаної пам'яті;
- 8) зазначення ресурсів ЕОМ для виконання модуля;
- 9) опис виняткових ситуацій та рекомендації щодо їх подолання;
- 10) список інших програм, які використовуються при виконанні модуля;
- 11) опис контрольного прикладу, вихідних даних та результатів розрахунку.

Наведені правила написання модулів є досить загальними і застосовуються для модулів, написаних для різних систем і різними мовами. У конкретних випадках вони можуть доповнюватися і уточнюватися. Паспорт може зберігатися разом з модулем як примітка чи у спеціальній бібліотеці.

1.5. Інтелектуальні методи в управлінні

Прагнення отримати точну, вичерпну модель для досить складного об'єкта (процесу) не має сенсу, оскільки складність такого опису стає сумірною зі складністю самого об'єкта. Використання такої моделі дозволяє наочно і просто пояснити механізм його функціонування і скористатися будь-якими стандартними математичними процедурами для дослідження характеристик об'єкта. Особливо це відноситься до таких об'єктів як виробничі процеси, організаційні, транспортні, біологічні, економічні системи та ін. Згідно з принципом Лотфі А. Заде [7] "чим складніша система, тим менше ми здатні дати точні і в той же час такі, що мають практичне значення, судження про її поведінку. Для систем, складність яких перевищує деякий пороговий рівень, точність і практичний сенс стають майже виключними одна одну характеристиками. У цьому сенсі точний якісний аналіз поведінки гуманістичних систем (систем, в яких бере участь людина) не має, мабуть, великого практичного значення в реальних соціальних, економічних та інших завданнях, пов'язаних з участю однієї людини або групи людей".

Так, широке застосування за останні десятиліття отримали зовсім недавно відомі лише вузькому колу фахівців нейронні мережі, нечітка логіка, генетичні алгоритми і ряд інших інформаційних технологій.

Під інтелектуальними методами розуміються такі способи розв'язання завдань, в основі яких лежать алгоритми та дії, більшою чи меншою мірою пов'язані з інтелектуальною діяльністю людини. Термін "інтелектуальні" в області комп'ютерних технологій можна вважати усталеним і поєднання "інтелектуальні інформаційні технології" не рідко слух і сприймається фахівцями однозначно [8]. Клас інтелектуальних технологій (ІТ) включає такі напрямки:

- штучні нейронні мережі (ШНМ);
- нечітка логіка (НЛ);

- генетичні алгоритми (ГА);
- нелінійна динаміка (НД).

Штучні нейронні мережі складаються з окремих обчислювальних елементів (формальних нейронів), які певною мірою подібні до біологічних нейронів мозку людини. Характерна особливість ШНС полягає в тому, що процес програмування традиційним шляхом вирішення завдання замінюється процедурою навчання мереж. Метод навчання ШНМ є одним з головних класифікаційних ознак мереж. Спосіб об'єднання нейронів у мережі, кількість шарів нейронів, наявність (відсутність) зворотних зв'язків визначають архітектуру ШНС. Мережі різних архітектур призначені для вирішення різних завдань, з яких найбільш важливий клас погано формалізованих завдань, де немає явних залежностей між вхідними і вихідними величинами. До проблем, які можуть бути вирішені з допомогою ШНМ, відносяться завдання класифікації і ранжирування підприємств, фірм, побудови рейтингів банків, прогнозування, зміни обмінного курсу валют і т.і. Нечітка логіка і правила, що засновані на її концепції, являють собою засіб моделювання невизначеностей природних понять мови. Прикладами таких невизначеностей служать лінгвістичні змінні «холодна», «тепла», «гаряча» стосовно до температури води. На вхід системи з нечіткою логікою поступають чіткі змінні, які перетворюються на нечіткі підмножини. З допомогою експертів створюється база правил виду "якщо ..., то ...", включається в систему. Над вхідними змінними за допомогою бази правил виконуються необхідні перетворення, після чого за допомогою методів дефазифікації (переходу від нечітких змінних до чітких) на виході системи формується чітка вихідна змінна. Серед управлінських завдань, розв'язуваних з допомогою систем НЛ, можна виділити клас проблем, де при нечітких вхідних змінних потрібно отримати кількісну характеристику вихідної величини.

Генетичні алгоритми являють собою алгоритми пошуку оптимальних рішень, які побудовані на принципах природного

відбору і генетики. Будь-яке можливе рішення відображається у вигляді рядка (хромосоми) фіксованої довжини, популяції яких застосовуються традиційні генетичні оператори: **селекція, схрещування, мутація**. **Селекція** направляє генетичний пошук у перспективні райони простору рішень, **схрещування** виконує випадковий пошук поблизу локального оптимуму, **мутація** відшукує покращене рішення. Визначаючи в кожному поколінні кращі хромосоми, схрещуємо їх між собою для отримання потомства, яке являє близьке до оптимального вирішення завдання. До задач, що розв'язуються за допомогою ГА, можна віднести складання плану оптимальних перевезень, визначення кращої торгової стратегії, розміщення виробничих потужностей.

Нелінійна динаміка або хаос узгоджується більшою мірою щоденним досвідом людини набагато сильніше, ніж точна передбачуваність. Нелінійна динаміка вивчає властивості нелінійних динамічних систем і використовує для опису систем нелінійні моделі, які зазвичай описуються диференціальними рівняннями і дискретними відображеннями.

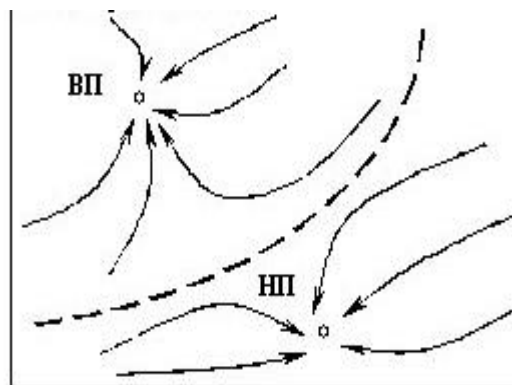
Теорія хаосу – математичний апарат, що описує поведінку деяких нелінійних динамічних систем, схильних, за певних умов, до явища, відомого як хаос (динамічний хаос, детермінований хаос). Поведінка такої системи здається випадковою, навіть якщо модель, що описує систему, є детермінованою.

Синергетика або **теорія складних систем** – міждисциплінарний напрям науки, що вивчає загальні закономірності явищ і процесів у складних нерівноважних системах (фізичних, хімічних, біологічних, екологічних, соціальних та інших) на основі властивих їм принципів самоорганізації. Синергетика є міждисциплінарним підходом, оскільки принципи, що керують процесами самоорганізації, подаються одними і тими ж незалежними від природи систем, і для їх опису повинен бути придатний загальний математичний апарат.

Дослідження економіки, як системи, що розвивається, показують, що всупереч широко поширеній думці про одиничність

ринкової рівноваги в ринковій економіці можливі кілька рівноважних станів (рис.1.11) з різною продуктивністю виробництва, з різним рівнем життя населення і ступенем соціального (майнового) розшарування (механізм виникнення кількох стійких станів описаний, наприклад, у [9]).

Обсяг виробництва



Майнове розшарування

Рис.1.11. Розрахункові траєкторії розвитку ринкової економіки:
ВП – високопродуктивний стан, НП – низькопродуктивний стан

В еволюційній економіці вони називаються інституціональними пастками, в теорії систем країн – атракторами. Дані стану при відсутності соціальних катаклізмів досить стійкі. Те, в якому з можливих станів опиниться економіка країни, залежить від вихідних умов, від дій уряду, ресурсних можливостей та інших чинників. Ринкові відносини утримують економіку в рівноважному стані, переходи між рівноважними станами можливі, але не спонтанно, а в результаті цілеспрямованих державних заходів.

Методи нелінійної динаміки дозволяють моделювати швидкі нерівноважні процеси (так звані «фазові переходи») в економічних системах, пов'язані з переходом від одних стійких станів до інших. Перехід з високопродуктивного в низькопродуктивний стійкий стан зовнішньо сприймається як економічна криза. Можна показати [10], що динаміка ВВП на душу населення в розвинених країнах і в світі в цілому добре апроксимується степеневією залежністю типу:

$$Y(t) = \frac{C}{(2026-t)^k}, \quad (1.4)$$

де – зміна ВВП на душу населення, C – константа, t – рік, k – показник ступеня (близько одиниці). Залежність (1.4) характеризує режим із загостренням, відповідно до якого, при збереженні існуючих тенденцій, у 2026 році ВВП на душу населення повинен перетворитися на нескінченність. Оскільки це фізично неможливо, реально це означає, що до 2026 року мають відбутися кардинальні зміни в світовій економіці. Економічна криза, що вибухнула, можливо, є першою ластівкою цього процесу.

Характерною особливістю сучасної ситуації є те, що економічно розвинені країни увійшли в режим коеволюційного розвитку (зміни одного виду призводять до змін в іншому): незважаючи на відмінність стартових умов, їх економічна динаміка характеризується близькими за значенням моментами загострення (в районі 2026 року), йде підтягування країн цієї групи до загального рівня (це добре видно на прикладі Тайваню, Гонконгу і Сінгапуру). Коеволюційні процеси у світовій економіці можуть бути описані досить простими нелінійними моделями [10] типу:

$$\begin{aligned} \frac{dL}{dt} &= a \cdot L - c \cdot (L - Y) \\ \frac{dY}{dt} &= b \cdot Y + c \cdot (L - Y) \\ L(0) &= L_0; Y(0) = Y_0, \end{aligned} \quad (1.5)$$

де L , Y – ВВП на душу населення двох взаємодіючих країн, c – коефіцієнт, що характеризує інтенсивність взаємодії, $L(0) = L_0; Y(0) = Y_0$ – початкові умови ($L_0 > Y_0$ тобто L – країна-лідер, Y – країна, що здоганяє).

Чисельне рішення системи (1.5) дає картину, близьку до спостережуваних статистичних даних. Результати моделювання

показують таке: розбіжності (єдиний момент загострення) виникають навіть при досить малих рівнях взаємодії систем; підтягування "до свого рівня" (зближення траєкторій розвитку: $Y(t) > L(t)$) вимагає великих витрат ресурсів з боку лідера L і слабо залежить від початкових умов $Y(0) = Y_0$ і режимів взаємодії систем; економічна допомога з метою непогіршення відносини $Y(t)/L(t)$ вимагає суттєво менших витрат ресурсів з боку лідера L і пропорційна відношенню $Y(0)/L(0)$ (досвід взаємодії розвинених країн Заходу і країн "третього світу").

Прогнози за допомогою традиційних моделей, заснованих на аналізі рівноважних станів, не заслуговують довіри. Необхідно переходити до моделей, націлених на опис нерівноважних процесів, що використовують апарат нелінійної динаміки, синергетики. Велике значення для діяльності світового фінансового ринку має завдання прогнозування його динаміки як у періоди зростання, так і в стані кризи і падіння. Ця задача важлива як для практиків, так і для економістів і фінансистів-теоретиків. Учасникам фінансових ринків необхідно оцінювати прибутковість вкладень, оптимізувати портфель, визначати моменти входу на ринок і виходу з нього, обчислювати справедливую вартість деривативів. Ризик-менеджерам потрібно точно оцінювати ризикові характеристики активів і портфелів, так як вартість під ризиком.

Для рішення подібних задач розроблено безліч різних підходів – від емпіричних методів технічного аналізу до агентно-орієнтованих імітаційних моделей або технологій «розкопування даних» (datamining). Таке різноманіття пояснюється тим, що динаміка прибутковості на фінансових ринках не може бути якісно описана лінійними моделями ARMA і випадкового блукання.

При цьому нелінійні методи аналізу динаміки і відповідні технології управління ризиками розроблені й успішно застосовуються для розвинених фінансових ринків США та Європи. Вони мало вивчені по відношенню до нестійким ринків.

РОЗДІЛ 2

НЕЙРОННІ МЕРЕЖІ

Завдання, які вирішуються інформаційними системами, у більшості випадків можна звести до ряду типових, серед яких можна виділити такі:

- 1) визначення майбутнього процесу за його минулим і теперішнім (прогноз);
- 2) знаходження рішень, які максимізують або мінімізують певний критерій якості при заданих обмеженнях (оптимізація);
- 3) переклад і підтримку системи в необхідному стані (управління);
- 4) оцінка невідомої залежності за експериментальними даними (апроксимація функцій);
- 5) визначення приналежності образу одному або декільком попередньо визначеним класам (класифікація образів).

Поняття "нейронні мережі" з'явилося в 40-х роках ХХ століття в середовищі нейробіологів і нейроанатомів, які вивчали організацію і функціонування мозку. У процесі вивчення з'ясувалося, що мозок має приголомшливу складність (майже 100 мільярдів нейронів і кілька трильйонів сполук).

Поглиблення уявлень про функціонування нейронів дало можливість дослідникам створювати різноманітні математичні моделі. З'явилися штучні нейронні мережі (НМ), що представляють собою набори елементарних нейроноподобних перетворювачів інформації (нейронів, з'єднаних один з одним каналами обміну інформацією для їх спільної роботи).

В якості визначення нейронних мереж може бути прийнято: ШНМ (штучна нейронна мережа) – паралельно розподілена структура обробки інформації, що складається з окремих елементів (нейронів), з'єднаних між собою зв'язками.

Комп'ютери, побудовані на основі нейронних мереж (нейро-комп'ютери) володіють цілим рядом властивостей, привабливих з точки зору їх практичного використання:

1. Надвисока продуктивність за рахунок використання масового паралелізму обробки інформації.

2. Толерантність до помилок: працездатність зберігається при пошкодженні значного числа нейронів.

3. Здатність до навчання; програмування обчислювальної системи замінюється навчанням.

4. Здатність до розпізнавання образів в умовах сильних перешкод і спотворень.

Однак, перші дві властивості мають місце тільки при апаратній реалізації нейронних мереж. Апаратно реалізовані нейронні мережі забезпечують вирішення складних завдань за час близько часу спрацьовування ланцюжків електронних та/або оптичних елементів. Рішення слабо залежить від несправності окремого нейрона. Це робить їх привабливими для використання в складі бортових обчислювальних систем.

Пакет Neural Networks Toolbox системи Matlab містить засоби для проектування, моделювання, навчання та використання множини відомих парадигм сучасного апарату штучних нейронних мереж: від базових моделей перцептрона до найсучасніших асоціативних і самоорганізуючих мереж.

У даний час ШНМ застосовуються для вирішення багатьох неформалізованих або важко формалізованих завдань [11–17]:

- розпізнавання і синтезу мови;
- розпізнавання аерокосмічних зображень;
- прогнозування котирування цінних паперів та курсу валют;
- попередження шахрайства з кредитними картками;
- оцінки вартості нерухомості;
- оцінки фінансового стану підприємств та ризику неповернення кредитів;
- обробки радіолокаційних сигналів;

- контролю руху на швидкісних автомагістралях і залізницях;
- діагностики в медицині;
- видобування знань з великих обсягів даних в бізнесі, фінансах і наукових дослідженнях.

Нейронні мережі можна використовувати за таких умов:

- Якщо завдання може вирішувати людина.
- Якщо при розв'язанні задачі можна виділити множину вхідних факторів (сигналів, ознак, даних тощо) і множину вихідних факторів.

• Якщо зміни вхідних факторів призводять до зміни вихідних. У той же час застосування нейронних мереж при вирішенні деяких завдань може виявитися ефективнішим використанням розуму людини. Це пояснюється тим, що людський розум орієнтований на вирішення завдань у тривимірному просторі. Багатовимірні завдання для нього характеризуються значно більшою трудомісткістю.

Штучним нейронним мережам не властиве таке обмеження. Їм все одно – вирішувати тривимірну або 10-вимірну задачу. Місце нейромережної технології серед інших методів обробки даних показано на рис. 2.1, 2.2.

Подібно до біологічної нейронної системи ШНМ є обчислювальною системою з величезним числом паралельно функціонуючих простих процесорів з безліччю зв'язків. Моделі ШНМ у деякій мірі відтворюють "організаційні" принципи, властиві мозку людини. Моделювання біологічної нейронної системи з використанням ШНМ може також сприяти кращому розумінню біологічних функцій. Такі технології виробництва, як VLSI (надвисокий рівень інтеграції) і оптичні апаратні засоби, роблять можливим подібне моделювання.

Глибоке вивчення ШНМ вимагає знання нейрофізіології, науки про пізнання, психології, фізики (статистичної механіки), теорії управління, теорії обчислень, проблем штучного інтелекту, статистики/математики, розпізнавання образів, комп'ютерного

зору, паралельних обчислень і апаратних засобів (цифрових/аналогових/VLSI /оптичних).

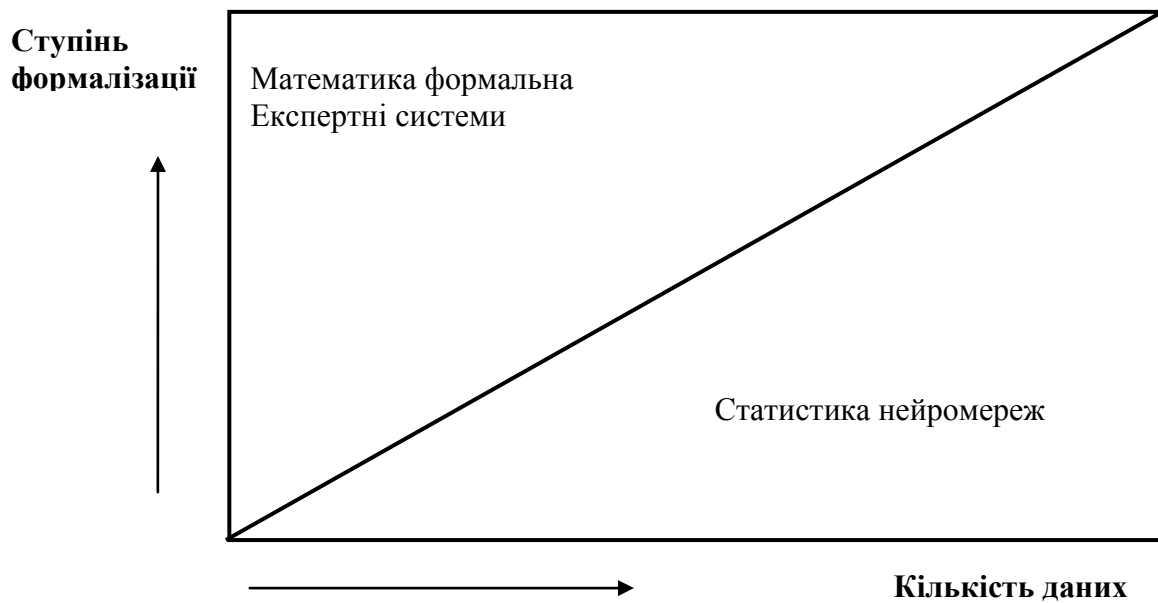


Рис. 2.1. Область використання нейронних мереж

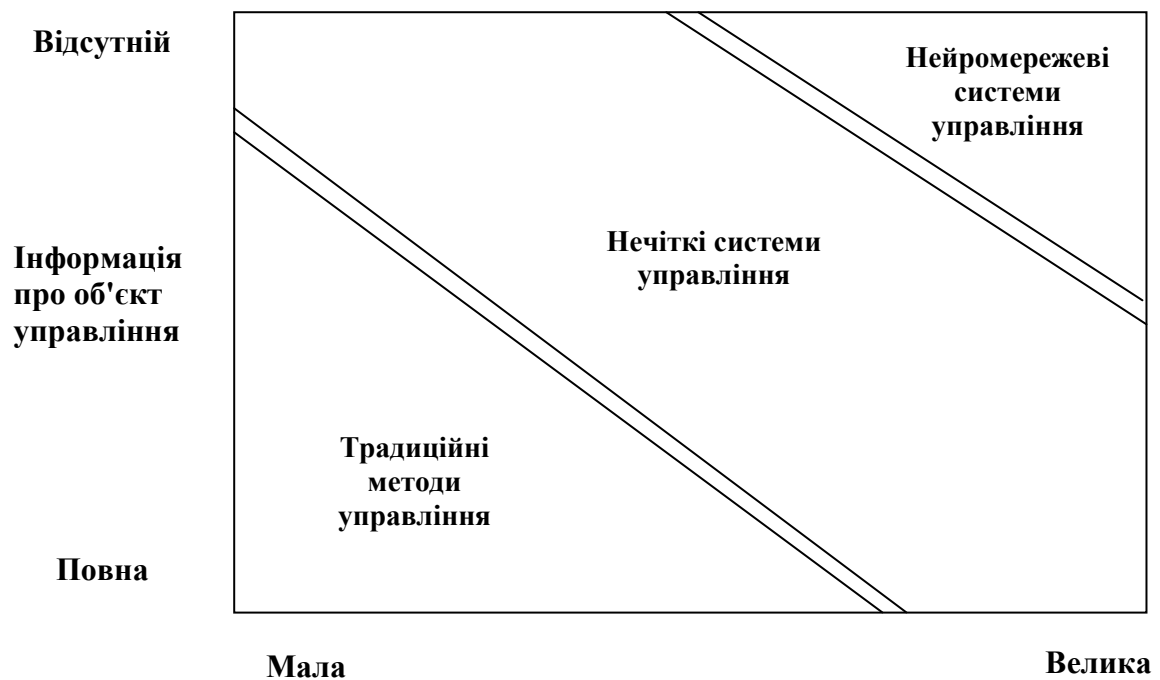


Рис.2.2. Области ефективного застосування різних систем управління

З іншого боку, ШНС також стимулюють ці дисципліни, забезпечуючи їх новими інструментами та уявленнями. Цей симбіоз життєво необхідний для досліджень нейронних мереж. Наведемо деякі проблеми, які вирішуються в контексті ШНС і представляють інтерес для вчених та інженерів.

Класифікація образів. Завдання полягає у зазначенні приналежності вхідного образу (наприклад, мовного сигналу або рукописного символу), представленого вектором ознак, одним або декількома попередньо визначеними класами. До відомих програм належать розпізнавання букв, розпізнавання мови, класифікація сигналу електрокардіограми, класифікація клітин крові.

Кластеризація/категоризація. При вирішенні задачі кластеризації, яка відома також як класифікація образів "без вчителя", відсутня навчальна вибірка з мітками класів. Алгоритм кластеризації заснований на подібності образів і розміщує близькі образи в один кластер. Відомі випадки застосування кластеризації для отримання знань, стиснення даних і дослідження властивостей даних.

Апроксимація функцій. Припустимо, що існує навчальна вибірка $((x_1, y_1), (x_2, y_2), \dots, (x_n, y_n))$ (пари даних вхід – вихід), яка генерується невідомою функцією $f(x)$, спотвореною шумом. Завдання апроксимації полягає в знаходженні оцінки невідомої функції $f(x)$. Апроксимація функцій необхідна при вирішенні численних інженерних і наукових задач моделювання.

Прогноз. Нехай задано n дискретних відліків $\{y(t_1), y(t_2), \dots, y(t_n)\}$ у послідовні моменти часу t_1, t_2, \dots, t_n . Завдання полягає в передбаченні значення $y(t_{n+1})$ в деякий майбутній момент часу t_{n+1} . Прогноз мають значний вплив на прийняття рішень в бізнесі, науці і техніці. Прогноз цін на фондовій біржі і прогноз погоди є типовими додатками техніки передбачення/прогнозу.

Оптимізація. Численні проблеми в математиці, статистиці, техніці, науці, медицині та економіці можуть розглядатися як проблеми оптимізації. Завданням алгоритму оптимізації є знаходження такого рішення, яке задовольняє системі обмежень і максимізує або мінімізує цільову функцію. Задача комівояжера, що відноситься до класу NP-повних, є класичним прикладом задачі оптимізації.

Управління. Розглянемо динамічну систему, задану сукупністю $\{u(t), y(t)\}$, де $u(t)$ є вхідним керуючим впливом, а $y(t)$ – виходом системи в момент часу t . В системах керування з еталонною моделлю метою управління є розрахунок такого вхідного впливу $u(t)$, при якому система слідує бажаною траєкторією, що диктується еталонною моделлю. Прикладом є оптимальне керування двигуном.

При застосуванні нейронних мереж необхідно вирішити такі завдання:

1. Постановка задачі, придатної для вирішення за допомогою нейронної мережі.
2. Вибір моделі ШНМ.
3. Підготовка вихідних даних для навчання ШНМ.
4. Навчання ШНМ.
5. Власне розв'язання задачі за допомогою навченої ШНМ.

Крім того, іноді потрібний ще один етап – інтерпретація рішення, отриманого нейронною мережею.

Найбільш трудомісткими процесами при використанні нейронних мереж є підготовка вихідних даних для навчання і навчання нейронної мережі.

Нервова система і мозок людини складаються з нейронів, з'єднаних між собою нервовими волокнами (рис.2.3). Нервові волокна здатні передавати електричні імпульси між нейронами. Всі процеси передачі подразнень від нашої шкіри, вух і очей до мозку, процеси мислення та управління діями – все це реалізовано в живому організмі як передача електричних імпульсів між нейро-

нами. Розглянемо будову біологічного нейрона. Кожен нейрон має відростки нервових волокон двох типів – дендрити, з яких приймаються імпульси, і єдиний аксон, за яким нейрон може передавати імпульс. Аксон контактує з дендритами інших нейронів через спеціальні утворення – синапси, які впливають на силу імпульсу.

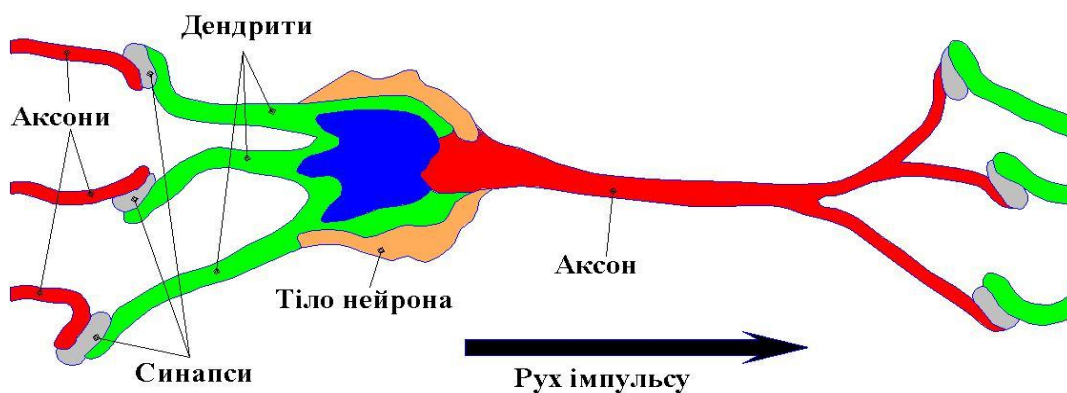


Рис. 2.3. Будова біологічного нейрона

Можна вважати, що при проходженні синапсу сила імпульсу змінюється у певне число раз, яке ми будемо називати вагою синапсу. Імпульси, що надійшли до нейрона одночасно за кількома дендритами, підсумовуються. Якщо сумарний імпульс перевищує деякий поріг, нейрон збуджується, формує власний імпульс і передає його далі по аксону. Передача інформації між нейронами здійснюється *електричними імпульсами*. Під дією цих сигналів виникає процес хімічної дифузії іонів натрію, калію, хлору і деяких інших елементів, змінює електричний *потенціал мембрани*. Коли цей потенціал (сумарний імпульс) досягає деякої величини, званої *порогом нейрона*, нейрон збуджується і виробляє імпульс, який йде по аксону. При цьому потенціал мембрани різко падає і нейрон "розряджається". Після деякої паузи нейрон може знову сформувати імпульс. Важливо відзначити, що ваги

синапсів можуть змінюватися з часом, а отже, змінюється і поведінка відповідного нейрона. Вихідний сигнал проходить по гілках аксона і досягає *синапсів*, які з'єднують аксони з дендритними деревами інших нейронів. Всі нейрони виробляють сигнали тільки одного знаку. Якщо імпульси, що надходять на синапс, призводять до підвищення мембранного потенціалу, то такий синапс називається *збудливим*. В іншому випадку синапс називається *гальмівним*. Величина вхідного сигналу, генерованого синапсом, може бути різною навіть при однаковій величині сигналу, що проходить через синапс. Ці відмінності визначаються ефективністю або *вагою синапса*. Важливо відзначити, що ваги синапсів можуть змінюватися з часом, а отже, змінюється і поведінка відповідного нейрона.

Під нейронними мережами маються на увазі обчислювальні структури, які моделюють прості біологічні процеси, зазвичай асоційовані з процесами людського мозку. Адаптуються і навчені, вони являють собою розпаралелені системи, здатні до навчання шляхом аналізу позитивних і негативних впливів. Елементарним перетворювачем в даних мережах є *штучний нейрон* або просто *нейрон*, названий так за аналогією з біологічним прототипом.

До теперішнього часу запропоновано і вивчено велику кількість моделей нейроноподібних елементів і нейронних мереж.

2.1. Структура штучного нейрона

Структура штучного нейрона показана на рис. 2.4. До складу нейрона входять помножувачі (синапси), суматор і нелінійний перетворювач.

Синапси здійснюють зв'язок між нейронами і множать вхідний сигнал на число, що характеризує силу зв'язку, – вага синапса. Суматор виконує додавання сигналів, що надходять по синаптичних зв'язках від інших нейронів, і зовнішніх вхідних сигналів.

Нелінійний перетворювач реалізує нелінійну функцію одного аргументу – виходу суматора. Ця функція називається *функцією активації* або *передатною функцією* нейрона.

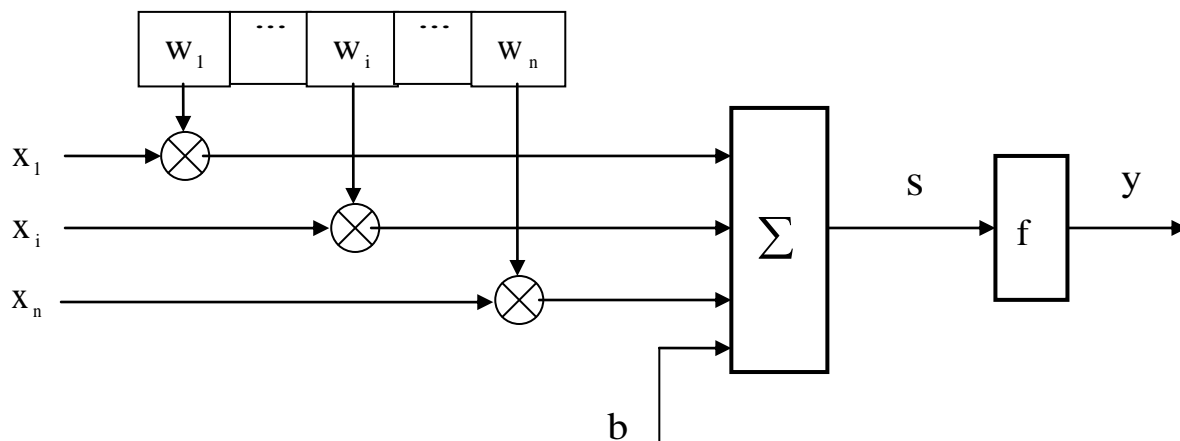


Рис.2.4. Структура штучного нейрона

Нейрон в цілому, реалізує скалярну функцію векторного аргументу. Незавжно побудувати математичну модель описаного процесу. Математична модель нейрона описується співвідношеннями:

$$s = \sum_{i=1}^n w_i x_i + b_i, \quad y = f(s),$$

де w_i – вага синапсу, $i = 1, \dots, n$; s – результат підсумовування; x_i – компонент вхідного вектора (вхідний сигнал); y – вихідний сигнал нейрона; n – число входів нейрона; f – нелінійне перетворення (функція активації або передатна функція); b_i – значення зміщення (у нейронну мережу іноді вводяться зміщення, яке діє як ваговий коефіцієнт від комірки з активацією, що дорівнює 1; зсув збільшує вхідний вплив мережу на одиницю; замість зміщення в ряді випадків застосовується фіксований поріг для функції активації).

2.2. Функції активації

У загальному випадку вхідний сигнал, вагові коефіцієнти та значення зсуву можуть набувати речових значень. Вихід визначається видом функції активації і може бути як дійсним, так і цілим.

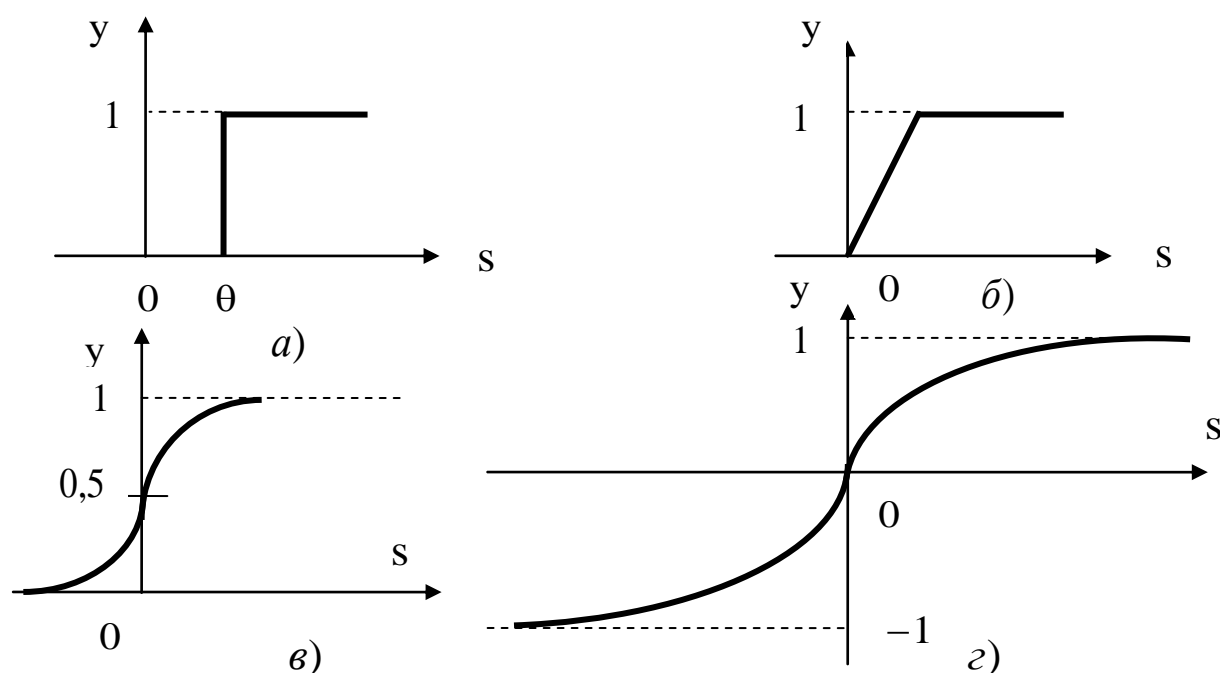


Рис. 2.5. Приклади активаційних функцій: а – порогова; б – напівлінійна з насиченням; в – сигмоїд (логістична); г – сигмоїд (гіперболічний тангенс)

Синаптичні ваги з позитивними вагами називають збудливими, з негативними – гальмівними.

Нейрон перетворює отриманий сумарний імпульс відповідно до деякої передатної функції $f(s)$. Таким чином, нейрон повністю описується своїми вагами w_i і передатною функцією $f(s)$. Отримавши набір чисел (вектор) x_i в якості входів, нейрон видає деяке число y .

На вхідний сигнал s нелінійний перетворювач відповідає вихідним сигналом $f(s)$, який представляє собою вихід нейрона. Приклади активаційних функцій представлені в табл.2.1 і на рис. 2.5.

Таблиця 2.1. Приклади активаційних функцій

Назва	Формула	Область значень
Порогова	$f(s) = \begin{cases} 0, & s < \theta \\ 1, & s \geq \theta \end{cases}$	(0,1)
Знакова	$f(s) = \begin{cases} 1, & s > 0 \\ -1, & s \leq 0 \end{cases}$	(-1,1)
Сигмоїдальна (логістична)	$f(s) = \frac{1}{1 + e^{-s}}$	(0,1)
Напівлінійна	$f(s) = \begin{cases} s, & s > 0 \\ 0, & s \leq 0 \end{cases}$	(0,∞)
Лінійна	$f(s) = s$	(-∞,+∞)
Радіальна базисна (гаусова)	$f(s) = \exp(-s^2)$	(0,1)
Напівлінійна з насиченням	$f(s) = \begin{cases} 0, & s \leq 0 \\ s, & 0 < s < 1 \\ 1, & s \geq 1 \end{cases}$	(0,1)
Лінійна з насиченням	$f(s) = \begin{cases} -1, & s \leq -1 \\ s, & -1 < s < 1 \\ 1, & s \geq 1 \end{cases}$	(-1,1)
Гіперболічний тангенс (сигмоїдальна)	$f(s) = \frac{e^s - e^{-s}}{e^s + e^{-s}}$	(-1,1)
Трикутна	$f(s) = \begin{cases} 1 - s , & s \leq 1 \\ 0, & s > 1 \end{cases}$	(0,1)

Однією з найбільш поширених є нелінійна функція з насиченням, так звана логістична функція або сигмоїд:

$$f(s) = \frac{1}{1 + e^{-s}}$$

Одна з цінних властивостей сигмоїдної функції – простий вираз для її похідної:

$$f'(s) = f(s)[1 - f(s)],$$

який використовується в деяких алгоритмах навчання.

2.3. Класифікація нейронних мереж

Штучна нейронна мережа (ШНМ) – це набір нейронів, з'єднаних між собою. Існує безліч нейронних мереж, які класифікуються за кількома ознаками (табл.2.2). Найбільше поширення одержали шаруваті мережі прямого поширення.

Як правило, передаточні функції всіх нейронів у мережі фіксовані, а ваги є параметрами мережі і не можуть змінюватись. Деякі входи нейронів помічені як зовнішні входи мережі, а деякі виходи – як зовнішні виходи мережі.

Подаючи будь-які числа на входи мережі, ми отримуємо якийсь набір чисел на виходах мережі. Таким чином, робота нейромережі полягає в перетворенні вхідного вектора X у вихідний вектор Y , при чому це перетворення задається вагами мережі.

Нейронні мережі розрізняють за структурою мережі (зв'язками між нейронами), особливостями моделі нейрона, особливостями навчання мережі. За структурою нейронні мережі можна розділити (рис.2.6) на неповнозв'язні (або шаруваті) і повнозв'язні, з випадковими та регулярними зв'язками, з симетричними і несиметричними зв'язками.

Неповнозв'язні нейронні мережі (описувані неповнозв'язним орієнтованим графом і зазвичай звані перцептронами),

підрозділяються на одношарові (найпростіші персептрони) і багатшарові, з прямими, перехресними і зворотними зв'язками. Класичним варіантом шаруватих мереж є мережі прямого поширення (рис.2.7).

Таблиця 2.2. Типи штучних нейронних мереж

Тип	Опис
За топологією	
Повнозв'язні	Кожен нейрон пов'язаний з іншим нейроном у мережі (через високу складність навчання не використовується).
Шаруваті	Нейрони розташовуються шарами, кожен нейрон наступного шару пов'язаний з нейронами попереднього. Є одношарові і багатшарові мережі.
За типом зв'язків	
Прямого поширення	Всі зв'язки між нейронами йдуть від виходів нейронів попереднього шару до входів нейронів наступного.
Рекурентні	Допускаються зв'язки виходів нейронів наступних шарів зі входами нейронів попередніх.
Щодо організації навчання	
З учителем	При навчанні використовуються навчальні вибірки, в яких визначено необхідні від мережі вихідні значення, такі мережі використовують для вирішення задач класифікації.
Без учителя	Нейронна мережа сама в процесі роботи виділяє класи об'єктів і відносить об'єкт до певного класу, такі мережі використовують для задач кластеризації.
За типом сигналу	
Двійкові	На вхід нейронних мереж подають лише нулі або одиниці.
Аналогові	Подаються на входи нейронів сигнали можуть бути довільними (речовими числами).
За типом структури	
Однорідна	Всі нейрони в нейронній мережі використовують одну функцію активації.
Неоднорідна	Нейрони в нейронній мережі мають різні функції активації.

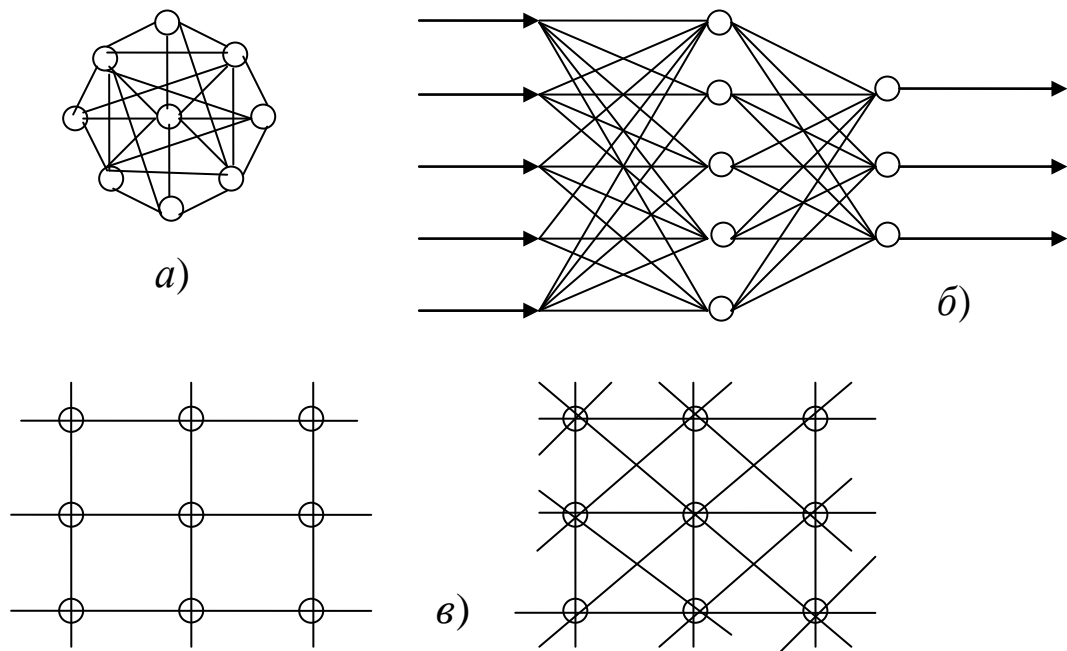


Рис.2.6. Архітектури нейронних мереж: а – повнозв'язна мережа; б – багатошарова мережа з послідовними зв'язками; в – слабозв'язні мережі

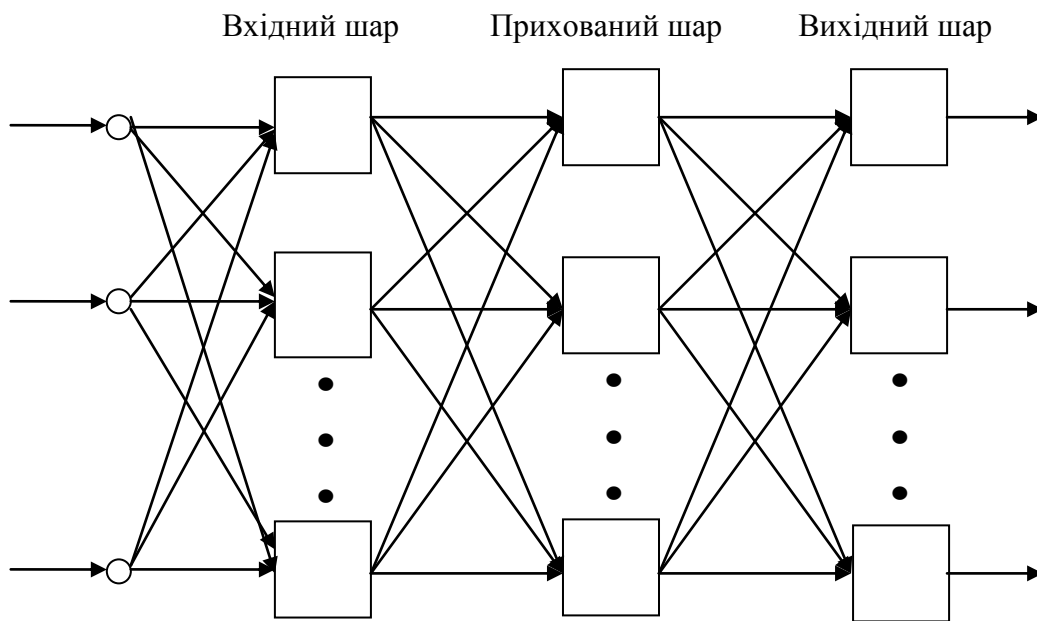


Рис.2.7. Багатошарова (двошарова) мережа прямого поширення

У нейронних мережах з прямими зв'язками нейрони j -го шару по входах можуть з'єднуватися тільки з нейронами i -их шарів, де $j > i$, тобто з нейронами нижчих шарів. У нейронних мережах з перехресними зв'язками допускаються зв'язки всередині одного шару, тобто вище наведене нерівність замінюється на $j \geq i$.

В нейронних мережах із зворотними зв'язками використовуються і зв'язки j -го шару по входах з i -им при $j < i$. Крім того, за видом зв'язків розрізняють персептрони з регулярними і випадковими зв'язками.

За способом подачі інформації на входи нейронної мережі розрізняють:

- подачу сигналів на синапси вхідних нейронів;
- подачу сигналів на виходи вхідних нейронів;
- подачу сигналів у вигляді ваг синапсів вхідних нейронів;
- адитивну подачу на синапси вхідних нейронів.

За способом зняття інформації з виходів нейронної мережі розрізняють:

- знімання з виходів вихідних нейронів;
- знімання з синапсів вихідних нейронів;
- знімання у вигляді значень ваг синапсів вихідних нейронів;
- адитивне знімання з синапсів вихідних нейронів.

Щодо організації навчання поділяють навчання нейронних мереж з учителем (supervised neural networks) і без вчителя (nonsupervised). При навчанні з учителем передбачається, що є зовнішнє середовище, яка надає навчальні приклади (значення входів і відповідні їм значення виходів) на етапі навчання або оцінює правильність функціонування нейронної мережі і відповідно зі своїми критеріями змінює стан нейронної мережі або заохочує (карає) нейронну мережу, запускаючи тим самим механізм зміни її стану.

Під станом нейронної мережі, яка може змінюватися, зазвичай розуміється:

- ваги синапсів нейронів (карта ваг –map) (коннекціоністський підхід);
- ваги синапсів і пороги нейронів (зазвичай в цьому випадку поріг є більш легко змінюваним параметром, ніж ваги синапсів);
- встановлення нових зв'язків між нейронами (властивість біологічних нейронів встановлювати нові зв'язки і ліквідувати старі називається пластичністю).

За способом навчання поділяють навчання за входами і виходами. При навчанні по входах повчальний приклад являє собою тільки вектор вхідних сигналів, а при навчанні за виходами у нього входить і вектор вихідних сигналів, відповідного вхідного вектора.

За способом пред'явлення прикладів розрізняють пред'явлення поодиноких прикладів і "сторінки" прикладів. У першому випадку зміна стану нейронної мережі (навчання) відбувається після пред'явлення кожного прикладу. У другому – після пред'явлення "сторінки" (множини) прикладів на основі аналізу їх всіх одразу.

Практично будь-яку задачу можна звести до задачі, розв'язуваної нейромережею.

У табл. 2.3 показано, яким чином слід сформулювати в термінах нейромережі завдання розпізнавання рукописних літер. Пояснимо, навіщо потрібно вибирати вихід з максимальним рівнем сигналу. Справа в тому, що рівень вихідного сигналу, як правило, може приймати будь-які значення з якогось відрізка. Однак, в даній задачі нас цікавить не аналогова відповідь, а лише номер категорії (номер букви в алфавіті). Тому використовується такий підхід - кожній категорії зіставляється свій вихід, а відповіддю мережі вважається та категорія, на якому виході максимальний рівень сигналу. У певному сенсі рівень сигналу на виході "А" – це вірогідність того, що на вхід була подана рукописна буква "А". Завдання, в яких потрібно віднести вхідні дані до однієї з відомих

категорій, називаються *завданнями класифікації*. Викладений підхід – стандартний спосіб класифікації за допомогою нейронних мереж.

Таблиця 2.3. Задача розпізнавання рукописних літер в термінах нейромережі

Задача розпізнавання рукописних літер	
Дано: растрове чорно-біле зображення літери розміром 30×30 пікселів	Треба: визначити, яка це літера (в алфавіті 33 букви)
Формулювання для нейромережі	
Дано: вхідний вектор з 900 двійкових символів (900=30×30)	Потрібно: побудувати нейромережу з 900 входами і 33 виходами, які позначені літерами. Якщо на вході мережі - зображення літери "А", то максимальне значення вихідного сигналу досягається на виході "А". Аналогічно мережа працює для всіх 33 літер.

Потім можна переходити до наступного питання – будівництво мережі. Це вирішується у два етапи:

- 1) вибір типу (архітектура мережі);
- 2) підбір ваг (навчання) мережі.

На першому етапі слід вибрати таке:

- які нейрони ми хочемо використовувати (число входів, передавальні функції);
- яким чином слід з'єднати їх між собою;
- що взяти в якості входів і виходів мережі.

Це завдання на перший погляд здається незрозумілим, але необов'язково вгадувати нейромережу "з нуля" – існує кілька десятків різних нейромережних архітектур, причому ефективність багатьох з них доведена математично. Найбільш поширені і вивчені архітектури – це багат шаровий перцептрон, нейромережа із загальною регресією, мережі Кохонена та інші. На **другому** етапі нам слід "навчити" обрану мережу, тобто підібрати такі значення

ваг, щоб мережа працювала належним чином. Ненавчена мережа подібна дитині – її можна навчити чому завгодно. У використовуваних на практиці нейромережах кількість ваг може становити кілька десятків тисяч, тому навчання – дійсно складний процес. Для багатьох архітектур розроблені спеціальні алгоритми навчання, які дозволяють налаштувати ваги мережі певним чином. Найбільш популярний з цих алгоритмів – метод зворотного поширення помилки (Error Back Propagation), що використовується, наприклад, для навчання персептрона.

Залежно від функцій, які виконуються нейронами в мережі, можна виділити три їх типи:

1) *вхідні* нейрони – на них подається вхідний вектор, що кодує вхідний вплив або образ зовнішнього середовища; обчислювальні процедури у цих нейронах не здійснюються, інформація передається зі входу на вихід шляхом зміни його активації;

2) *вихідні* нейрони – вихідні значення яких представляють вихід мережі;

3) *проміжні* нейрони – складають основу ШНМ, перетворення в них виконуються за вищенаведеними формулами.

Вибір структури ШНМ здійснюється відповідно до особливостей і складності завдання. Для вирішення деяких окремих типів задач існують оптимальні конфігурації. Якщо ж задача не може бути зведена ні до одного з відомих типів, розробнику доводиться вирішувати проблему синтезу нової конфігурації, при цьому необхідно керуватися такими принципами:

1) можливості мережі зростають зі збільшенням числа елементів мережі, щільності зв'язку між ними і числом виділених шарів;

2) уведення зворотних зв'язків поряд зі збільшенням можливостей мережі піднімає питання про динамічну стійкість мережі;

3) складність алгоритмів функціонування мережі (наприклад, уведення декількох типів синапсів) також сприяє посиленню можливостей мережі.

У більшості випадків оптимальний варіант виходить на основі інтуїтивного підбору, хоча в літературі наведені докази того, що для будь-якого алгоритму існує нейронна мережа, яка може його реалізувати.

Багато задач: розпізнавання образів (зорових, мовних), управління, прогнозування, ідентифікації складних систем, зводяться до такої постановки: *необхідно побудувати відображення таке, щоб у відповідь на кожен можливий вхідний сигнал X формувалася правильний вихідний сигнал Y* . Відображення задається кінцевим набором пар («вхід», «відомий вихід»). Число таких пар (навчальних прикладів) істотно менше загального числа можливих комбінацій значень вхідних та вихідних сигналів. Сукупність усіх навчальних прикладів носить назву навчальної вибірки.

У завданнях **розпізнавання** образів X – деяке уявлення образу (зображення, вектор чисел), Y – номер класу, до якого належить вхідний образ.

У задачах **управління** X – набір контрольованих параметрів керованого об'єкта, Y – код, що визначає керуючий вплив, що відповідає поточним значенням контрольованих параметрів.

У задачах **прогнозування** в якості вхідних сигналів використовуються тимчасові ряди, що представляють значення контрольованих змінних на деякому інтервалі часу. Вихідний сигнал – множина змінних, яка є підмножиною змінних вхідного сигналу.

Взагалі кажучи, велика частина прикладних задач може бути зведена до реалізації деякого складного багатовимірного функціонального перетворення $X \rightarrow Y$.

У результаті побудови такого перетворення (відображення) необхідно домагатися того, щоб забезпечувалося формування правильних вихідних сигналів відповідно:

- 1) з усіма прикладами навчальної вибірки;
- 2) з усіма можливими вхідними сигналами, які не увійшли до навчальної вибірки.

Теоретичною основою для побудови НМ є таке твердження: для будь-якої множини пар вхідних-вихідних векторів довільної розмірності існує двошарова однорідна нейронна мережа з послідовними зв'язками, з сигмоїдальними передавальними функціями і з кінцевим числом нейронів, що для кожного вхідного вектора X^k формує відповідний йому вихідний вектор Y^k .

Таким чином, для уявлення багатовимірних функцій багатьох змінних може бути використана однорідна нейронна мережа, що має всього один прихований шар, з сигмоїдальними передавальними функціями нейронів.

Для оцінки числа нейронів у прихованих шарах однорідних нейронних мереж можна скористатися формулою для оцінки необхідної кількості синаптичних ваг L_w (у багатошаровій мережі з сигмоїдальними передавальними функціями):

$$\frac{mN}{1 + \log_2 N} \leq L_w \leq m \left(\frac{N}{m} + 1 \right) (n + m + 1) + m,$$

де n – розмірність вхідного сигналу, m – розмірність вихідного сигналу, N – число елементів навчальної вибірки.

Число нейронів у двошаровій мережі складе: $L = \frac{L_w}{n + m}$ або

$$\frac{N}{10} - n - m \leq L \leq \frac{N}{2} - n - m.$$

Теорема про повноту. *Будь-яка безперервна функція на замкненій обмеженій множині може бути рівномірно наближена функціями, обчислюваними нейронними мережами, якщо функція активації нейрона двічі неперервно диференційовна і нелінійна.*

Таким чином, НМ є універсальними апроксимуючими системами. Задавшись певною структурою ШНМ, що відповідає будь-якій задачі, розробник мережі повинен знайти оптимальні значення всіх вагових коефіцієнтів. Цей етап називається навчанням ШНМ.

Таким чином, для вирішення конкретної задачі потрібно вибрати відповідну нейронну мережу. При цьому потрібно врахувати не тільки наведені в таблиці критерії, але й архітектуру мережі. Вибір архітектури передбачає визначення кількості шарів і нейронів у цих шарах. Не існує формального алгоритму визначення потрібної архітектури, тому на практиці вибирають або свідомо маленьку мережу і поступово її нарощують або свідомо велику і поступово виявляють невикористані зв'язки і скорочують мережу.

Нейронна мережа, перш ніж використовуватися на практиці для вирішення якої-небудь завдання, повинна бути навчена. Навчання нейронної мережі – це процес налаштування синаптичних ваг. Існує безліч алгоритмів, орієнтованих на певні типи мереж і на конкретні завдання, розглянемо алгоритми для одношарової і багатошарової мереж.

У загальному випадку завдання навчання НМ зводиться до знаходження деякої функціональної залежності $Y=F(X)$, де X – вхідний, а Y – вихідний вектори. Така задача, при обмеженому наборі вхідних даних, має нескінченну множину рішень. Для обмеження простору пошуку при навчанні ставиться задача мінімізації цільової функції помилки НМ, яка знаходиться за методом найменших квадратів:

$$E(W) = \frac{1}{2} \sum_{j=1}^p (y_j - d_j)^2, \quad (2.1)$$

де y_j – значення j -го виходу нейромережі; d_j – цільове значення j -го виходу; p – число нейронів у вихідному шарі.

Навчання нейромережі проводиться методом градієнтного спуску, тобто на кожній ітерації зміна ваги проводиться за формулою

$$\Delta W_{ij} = -\eta \cdot \frac{\partial E}{\partial W_{ij}}, \quad (2.2)$$

де η – параметр, що визначає швидкість навчання.

$$\frac{\partial E}{\partial W_{ij}} = \frac{\partial E}{\partial y_j} \cdot \frac{dy_j}{dS_j} \cdot \frac{\partial S_j}{\partial W_{ij}}, \quad (2.3)$$

де y_j – значення виходу j -го нейрона; S_j – зважена сума вхідних сигналів, визначається за формулою (2.1).

При цьому множник

$$\frac{\partial S_j}{\partial W_{ij}} \equiv x_i \quad (2.4)$$

де x_i – значення i -го входу нейрона.

Далі розглянемо визначення першого множника формули (2.3)

$$\frac{\partial E}{\partial y_j} = \sum_k \frac{\partial E}{\partial y_k} \cdot \frac{dy_k}{dS_k} \cdot \frac{\partial S_k}{\partial y_j} = \sum_k \frac{\partial E}{\partial y_k} \cdot \frac{dy_k}{dS_k} \cdot W_{jk}^{(n+1)}, \quad (2.5)$$

де k – число нейронів у шарі $n+1$.

Введемо допоміжну змінну

$$\delta_j^{(n)} = \frac{\partial E}{\partial y_j} \cdot \frac{dy_j}{dS_j} \quad (2.6)$$

Тоді ми зможемо визначити рекурсивну формулу для визначення n -го шару, якщо нам відомий наступний $n+1$ шар.

$$\delta_j^{(n)} = \left[\sum_k \delta_k^{(n+1)} \cdot W_{jk}^{(n+1)} \right] \cdot \frac{dy_j}{dS_j} \quad (2.7)$$

Знаходження ж для останнього шару НМ не є важким, так як нам відомий цільовий вектор, тобто вектор тих значень, які повинна видавати НМ при даному наборі вхідних значень.

$$\delta_j^{(N)} = (y_i^{(N)} - d_i) \cdot \frac{dy_i}{dS_i} \quad (2.8)$$

Запишемо формулу (2.2) у розкритому вигляді

$$\Delta W_{ij}^{(n)} = -\eta \cdot \delta_j^{(n)} \cdot x_i^n \quad (2.9)$$

Розглянемо тепер повний алгоритм навчання нейромережі:

1. Подати на вхід НС один з потрібних образів і визначити значення виходів нейронів нейромережі.

2. Розрахувати для вихідного шару НС за формулою (2.8) і розрахувати зміни ваг вихідного шару N за формулою (2.9).

3. Розрахувати за формулами (2.7) і (2.9) відповідно і $\Delta W_{ij}^{(N)}$ для інших шарів НМ, $n = N - 1, \dots, 1$.

4. Скорегувати всі ваги НС

$$\Delta W_{ij}^{(n)}(t) = W_{ij}^{(n)}(t-1) + \Delta W_{ij}^{(n)}(t). \quad (2.10)$$

5. Якщо істотна помилка, то перейти на крок 1.

На етапі 2 мережі почергово у випадковому порядку пред'являються вектори з навчальної послідовності.

2.3.1. Навчання нейронної мережі

Таким чином, нейромережі складені з простих елементів, що діють паралельно. Можна навчити нейромережу, регулюючи значення ваг між елементами. Зазвичай мережа регулюється або навчається так, щоб приватний вхід вів до цільового виходу. Регульована мережа заснована на порівнянні виходу і цілі, поки мережний вихід не буде відповідати меті. Звичайно є багато навчальних пар вхід/мета. Кількість навчальних пар вибирається залежно від постановки задачі. Змінні, які виступають у ролі вхідних і вихідних сигналів НС, можуть представляти собою експериментальні дані, отримані виміром вихідних параметрів об'єкта при завданні певних вхідних.

Навчити нейронну мережу – значить, повідомити їй, чого ми від неї домагаємося. Цей процес дуже схожий на навчання дитини алфавіту. Показавши дитині зображення літери "А", ми питає-

мо його: "Яка це літера?" Якщо відповідь невірна, ми повідомляємо дитині ту відповідь, який ми хотіли від неї отримати: "літера А". Дитина запам'ятовує цей приклад разом з вірною відповіддю, тобто в його пам'яті відбуваються деякі зміни в потрібному напрямку. Ми будемо повторювати процес пред'явлення букв знову і знову до тих пір, коли всі 33 літери будуть твердо запам'ятовані. Такий процес називають "**навчання з учителем**" (рис.2.8).

При навчанні мережі ми діємо абсолютно аналогічно. У нас є деяка база даних, що містить приклади (набір рукописних зображень літер). Пред'являючи зображення літери "А" на вхід мережі, ми отримуємо від неї деяку відповідь, не обов'язково вірну. Нам відома і вірна (бажана) відповідь – у даному випадку нам хотілося б, щоб на виході з міткою "А" рівень сигналу був максимальний. Зазвичай в якості бажаного виходу у задачі класифікації беруть набір (1, 0, 0, ...), де 1 стоїть на виході з міткою "А", а 0 – на всіх інших виходах. Обчислюючи різницю між бажаною відповіддю і реальною відповіддю мережі, ми отримуємо 33 числа – *вектор помилки*.

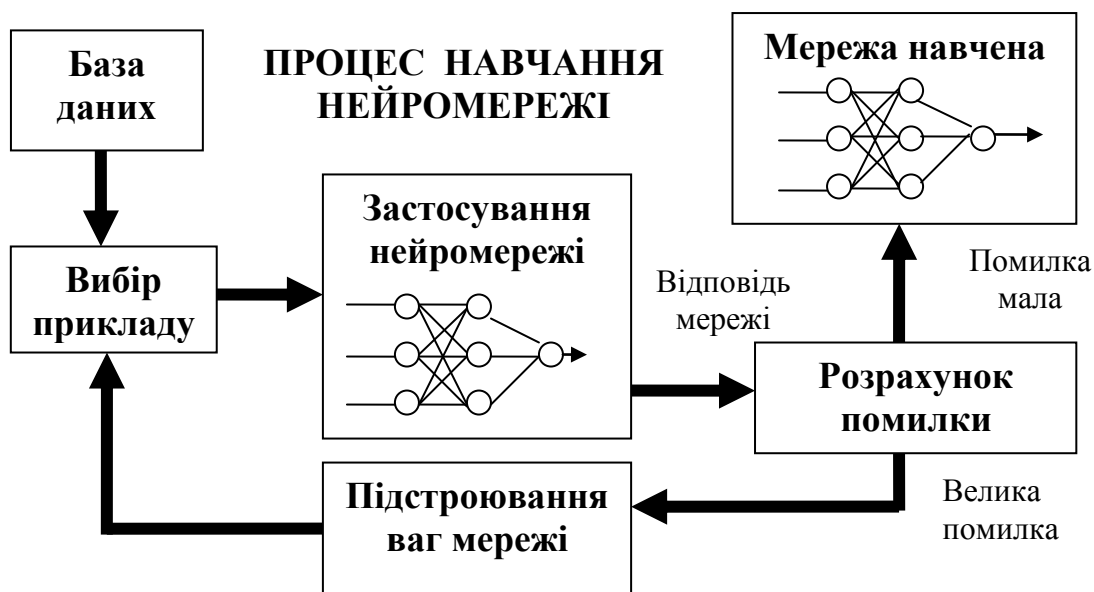


Рис.2.8. Ілюстрація процесу навчання НС

Алгоритм зворотного поширення помилки – це набір формул, який дозволяє за вектором помилки обчислити необхідні поправки для ваг мережі. Одну і ту ж літеру (а також різні зображення однієї і тієї ж букви) ми можемо пред'являти мережі багато разів. У цьому сенсі навчання швидше нагадує повторення вправ у спорті – тренування.

Виявляється, що після багаторазового пред'явлення прикладів ваги мережі стабілізуються, причому мережа дає правильні відповіді на всі (або майже всі) приклади з бази даних. У такому випадку кажуть, що "мережа вивчила всі приклади", "мережа навчена", або "мережа натренована". У програмних реалізаціях можна бачити, що у процесі навчання величина помилки (сума квадратів помилок на всіх виходах) поступово зменшується. Коли величина помилки досягає нуля або прийняттого малого рівня, тренування зупиняють, а отриману мережу вважають натренованою і готовою до застосування на нових даних. Важливо відзначити, що вся інформація, яку мережа має про завдання, міститься в наборі прикладів. Тому якість навчання мережі безпосередньо залежить від кількості прикладів у навчальній вибірці, а також від того, наскільки повно ці приклади описують дану задачу. Так, наприклад, безглуздо використовувати мережу для прогнозування фінансової кризи, якщо в навчальній вибірці криз не представлено. Вважається, що для повноцінного тренування потрібно хоча б кілька десятків (а краще сотень) прикладів.

Повторимо ще раз, що навчання мережі – складний і наукомісткий процес. Алгоритми навчання мають різні параметри і налаштування для керування якими потрібне розуміння їх впливу.

2.3.2. Алгоритми навчання нейронних мереж

Розглянемо ідею одного з найпоширеніших алгоритмів навчання – *алгоритму зворотного поширення помилки*. Це ітеративний градієнтний алгоритм навчання, який використовується з метою мінімізації середньоквадратичного відхилення поточного

виходу від бажаного виходу у багатошарових нейронних мережах. Цей алгоритм використовується для навчання багатошарових НМ з послідовними зв'язками виду (рис.2.9). Нейрони в таких мережах діляться на групи з загальним вхідним сигналом – шари. На кожний нейрон першого шару подаються всі елементи зовнішнього вхідного сигналу. Всі виходи нейронів m -го шару подаються на кожен нейрон шару $m+1$. Нейрони виконують зважене підсумовування елементів вхідних сигналів. До суми елементів вхідних сигналів, помножених на відповідні синаптичні ваги, додається зсув нейрона. Над результатом підсумовування виконується нелінійне перетворення – функція активації (передатна функція). Значення функції активації є вихід нейрона.

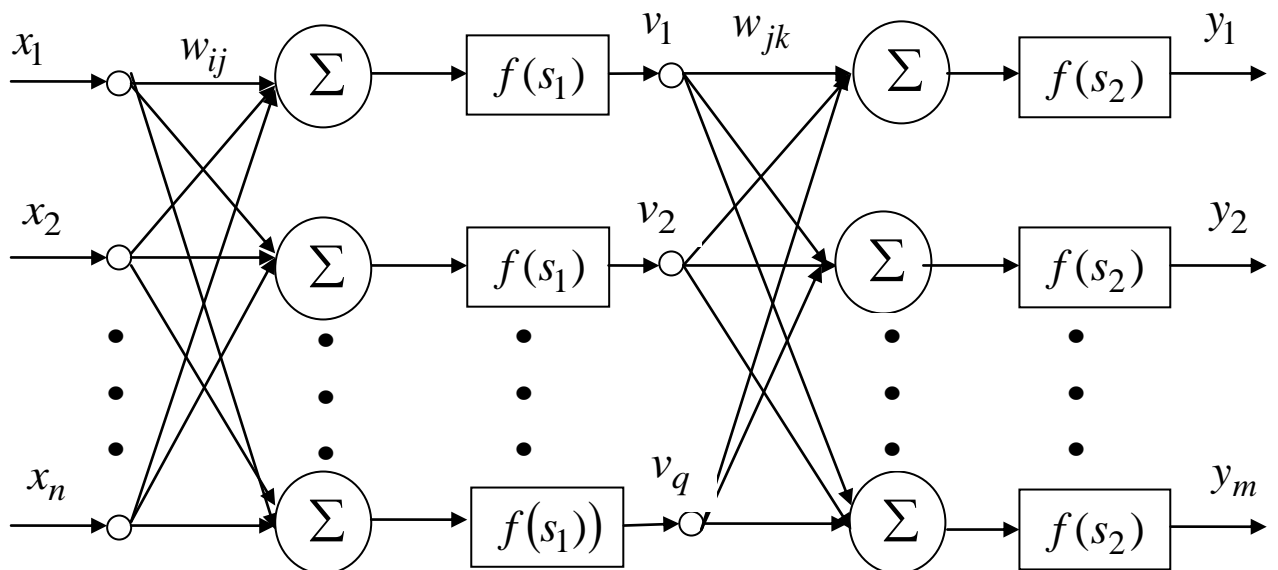


Рис. 2.9. Схема тришарового персептрона

У багатошарових мережах оптимальні вихідні значення нейронів усіх шарів, крім останнього, як правило, невідомі, і трьох- або більше шаровий персептрон вже неможливо навчити, керуючись тільки величинами помилок на виходах НМ. Найбільш прийнятним варіантом навчання в таких умовах виявився

градієнтний метод пошуку мінімуму функції помилки з аналізом сигналів помилки від виходів НМ до її входів, тобто в напрямі, зворотному прямому поширенню сигналів у звичайному режимі роботи. Цей алгоритм навчання НМ отримав назву процедури зворотного поширення.

У даному алгоритмі функції помилки являють собою суму квадратів неузгодженості (помилки) бажаного виходу мережі і реального. При обчисленні елементів вектора градієнта використаний своєрідний вигляд похідних функцій активації сигмоїдного типу. Алгоритм діє циклічно і його цикли зазвичай називають епохами. На кожному етапі на вхід мережі по черзі подаються всі навчальні спостереження, вихідні значення мережі порівнюються з цільовими значеннями й обчислюється помилка. Значення помилки, а також градієнта поверхні помилок використовується для коригування ваг, після чого всі дії повторюються. Початкова конфігурація мережі вибирається випадковим чином, і процес навчання припиняється або коли пройдено певну кількість епох, або коли помилка досягне деякого певного рівня малості, або коли помилка перестане зменшуватися.

Навчання мережі методом ЗПП включає в себе три етапи:

1. Пряме поширення вхідного навчального способу.
2. Обчислення помилки та її зворотне розповсюдження.
3. Регулювання ваг.

Цей метод заснований на обчисленні вектора градієнта поверхні помилок, який вказує напрямом найкоротшого спуску по поверхні з даної точки. Послідовність кроків приводить після ряду ітерацій до мінімуму поверхні помилок. Основна трудність тут є вибір довжини кроку. На практиці величина кроку приймається пропорційною крутизни схилу з постійною, так званою швидкістю навчання. Розглянемо алгоритм для багат шарового прямонаправленого персептрона (БПП) (рис.2.9).

Позначимо через $x_i (i = 1, 2, \dots, n)$ – вхідні нейрони; $v_j (j = 1, 2, \dots, q)$ – нейрони прихованого шару; $y_k (k = 1, 2, \dots, m)$ – вихідні нейрони; w_{ij} – ваги від осередків вхідного шару до нейронів прихованого шару; від ваги нейронів прихованого шару до вихідних нейронів. Індексом p ($p = 1, 2, \dots, P$) позначимо різні образи, що пред'являються на вхід. Вхідні сигнали можуть бути бінарними, біполярними або безперервними.

Поведінка мережі визначається на основі ряду пар «вхід-вихід». Кожен навчальний приклад складається з n вхідних сигналів x_i і m необхідних вихідних сигналів t_k . Навчання БПП для конкретної задачі еквівалентно знаходженню таких значень всіх синаптичних ваг, при яких для відповідного входу формується необхідний вихід або навчання багатошарової мережі полягає в регулюванні всіх ваг таким чином, щоб помилка між необхідними вихідними t_k і дійсними вихідними сигналами y_k , усереднена по всім навчальним прикладам, була мінімальна. При пред'явленні способу p на вхід мережі прихований осередок j приймає сигнал:

$$s_j^p = \sum_i w_{ij} x_i^p$$

і на своєму виході з допомогою функції активації виробляє такий сигнал:

$$v_j^p = f(s_j^p) = f\left(\sum_i w_{ij} x_i^p\right).$$

Вихідна клітинка з номером підсумовує сигнали від нейронів прихованого шару, утворюючи сигнал, рівний:

$$s_k^p = \sum_j w_{jk} v_j^p = \sum_j w_{jk} f\left(\sum_i w_{ij} x_i^p\right),$$

і після впливу функції активації формує вихідний сигнал (порогові значення відкинуті, оскільки вони завжди можуть бути введені в мережу):

$$y_k^p = f(s_k^p) = f\left(\sum_j w_{jk} v_j^p\right) = f\left[\sum_j w_{jk} f\left(\sum_i w_{ij} x_i^p\right)\right].$$

В якості функції помилок прийmemo функцію виду:

$$E[w] = 0,5 \sum_p \sum_k (t_k^p - y_k^p)^2,$$

де t_k^p – необхідне значення виходу.

Підставляючи в останній вираз значення, отримаємо:

$$E[w] = 0,5 \sum_p \sum_k \left\{ t_k^p - f\left[\sum_j w_{jk} f\left(\sum_i w_{ij} x_i^p\right)\right] \right\}^2.$$

Функція $E[w]$ є безперервно диференційовною функцією від кожної ваги, що входить до цього виразу, тому можна скористатися алгоритмом градієнтного спуску для знаходження ваг.

Для ваг між прихованим і вихідними шарами правило градієнтного спуску дає:

$$\Delta w_{jk} = -\eta \frac{\partial E}{\partial w_{jk}} = \eta \sum_p (t_k^p - y_k^p) \cdot f'(s_k^p) \cdot v_j^p = \eta \sum_p \delta_k^p v_j^p,$$

де $\delta_k^p = f'(s_k^p)(t_k^p - y_k^p)$, η – коефіцієнт швидкості навчання ($0 < \eta < 1$).

Для ваг між вхідним і прихованим шарами потрібно продиференціювати вираз для $E[w]$, скориставшись ланцюговим правилом:

$$\begin{aligned} \Delta w_{ij} &= \eta \frac{\partial E}{\partial w_{ij}} = -\eta \sum_p \frac{\partial E}{\partial v_j^p} \cdot \frac{\partial v_j^p}{\partial w_{ij}} = \eta \sum_p \sum_k (t_k^p - y_k^p) f'(s_k^p) w_{jk} f'(s_j^p) x_i^p = \\ &= \eta \sum_p \sum_k \delta_k^p w_{jk} f'(s_j^p) x_i^p = \sum_p \delta_j^p x_i^p, \end{aligned}$$

де $\delta_j^p = f'(s_j^p) \sum_k w_{jk} \delta_k^p$.

Зазначимо, що рівняння для Δw_{jk} і Δw_{ij} мають однакову форму, але розрізняються значеннями параметра δ . У загальному випадку при довільному числі шарів правило зміни вагу методі ЗПП має вигляд:

$$\Delta w_{\alpha\beta} = \eta \sum_p \delta_{out} v_{in},$$

де підсумовування проводиться по всім пропонованим образам; вихід і вхід відносяться до двох кінців α і β синаптичного з'єднання; v_{in} – активація від прихованої клітинки або реального входу. Значення δ залежать від розглянутого шару; для останнього шару ця величина визначається за виразом

$$\delta_k^p = f'(s_k^p)(t_k^p - y_k^p)$$

а для інших шарів – подібною формулою

$$\delta_j^p = f'(s_j^p) \sum_k w_{jk} d_k^p.$$

Вирази, що визначають правила зміни ваг, записані у вигляді сум за пропонованим образом, проте зазвичай образи надходять на вхід послідовно: образ p пред'являється на вхід, і по закінченні проходу «вперед-назад» по мережі всі ваги змінюються перед пред'явленням наступного способу. Це зменшує функцію помилок E на кожному кроці. Якщо образи вибираються у випадковому порядку, то рух по простору ваг відбувається стохастично, що дозволяє більш широко досліджувати поверхню помилок. Альтернативна версія зміни ваг (групове навчання) полягає у мінімізації функції помилки таким чином, що вагові зміни накопичуються по всіх навчальних прикладах і тільки потім відбувається модифікація ваг. Цей алгоритм являє собою таку послідовність кроків.

1. Ініціювати ваги, прийнявши їх малими випадковими величинами.

2. Якщо умова зупинки не виконується, робити кроки 3–10.

3. Вибирається чергова навчальна пара (X, Y) з навчальної множини; вектор X подається на вхід мережі. Для кожної навчальної пари виконувати кроки 4–9.

Прямий прохід по мережі

4. Кожний вхідний осередок приймає вхідний сигнал і поширює його до всіх нейронів прихованого шару.

5. Кожна комірka прихованого шару $v_j, j = 1, 2, \dots, q$ підсумовує свої зважені вхідні сигнали $s_j = \sum w_{ij}x_i$ застосовує до отриманої суми функцію активації, формуючи вихідний сигнал $v_j = f(s_j)$, який надсилається до всіх осередків вихідного шару.

6. Кожна вихідна комірka $y_k, k = 1, 2, \dots, m$ підсумовує зважені сигнали:

$$s_k = \sum w_{jk}v_j,$$

формуючи після застосування функції активації вихідний сигнал мережі:

$$y_k = f(s_k).$$

Зворотнє поширення помилки

7. Кожна вихідна комірka зіставляє своє значення виходу з необхідною цільовою величиною і обчислює параметр δ_k :

$$\delta_k = (t_k - y_k)f'(s_k)$$

Після чого визначається корективний член для терезів:

$$\Delta w_{jk} = \eta \delta_k v_j,$$

а параметри δ_k надсилаються в нейрони прихованого шару.

8. Кожна прихована комірka v_j підсумовує свої δ – входи від нейронів вихідного шару:

$$s_j = \sum_k \delta_k w_{jk}$$

Результат множиться на похідну від функції активації для визначення δ_j :

$$\delta_j = f'(s_j) \sum_k \delta_k w_{jk},$$

і обчислюється поправочний член: $\Delta w_{ij} = \eta \delta_j x_i$.

Зміна ваг

9. Ваги між прихованим і вихідними шарами модифікуються таким чином:

$$w_{jk}(new) = w_{jk}(old) + \Delta w_{jk}$$

Аналогічним чином змінюються ваги між вхідним і прихованими шарами:

$$w_{ij}(new) = w_{ij}(old) + \Delta w_{ij}.$$

10. Перевірка умови зупинки: мінімізація помилки між необхідним і реальним вихідними сигналами.

Схема алгоритму ЗПП представлена на рис. 2.10.

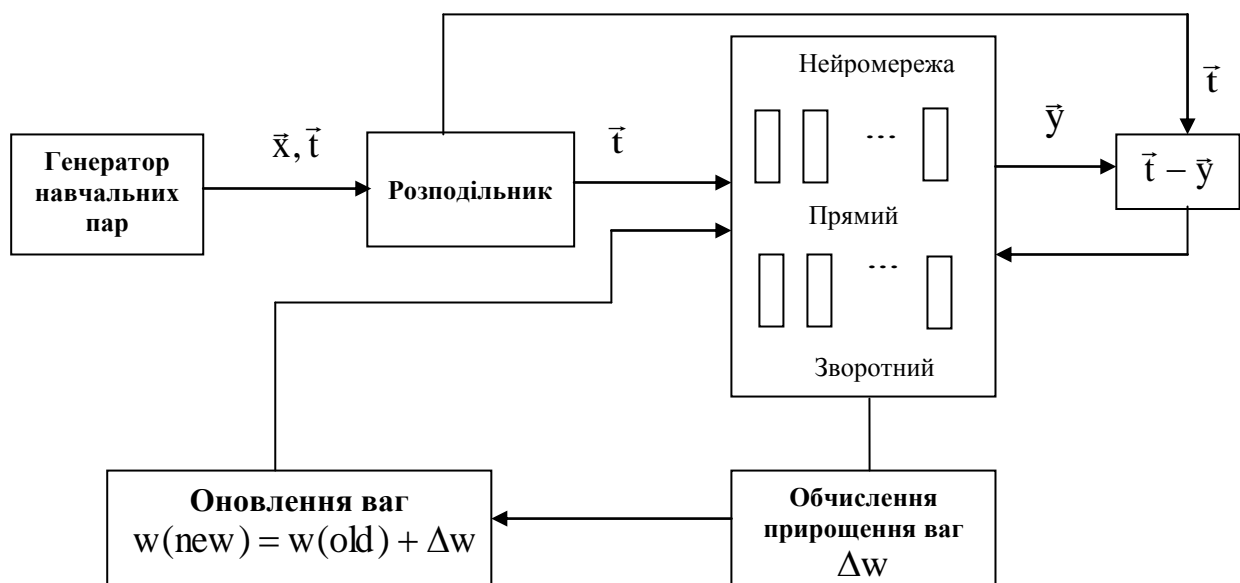


Рис.2.10. Обчислювальна схема методу ЗПП

2.3.2.1. Алгоритм навчання мережі за Δ -правилом

Найпростіша нейронна мережа – одношарова (рис. 2.11), що є розташованими паралельно нейронами, які отримують на входи однакові сигнали, але мають різні синаптичні зв'язки. Кількість входів і виходів такої нейронної мережі відповідає кількості нейронів.

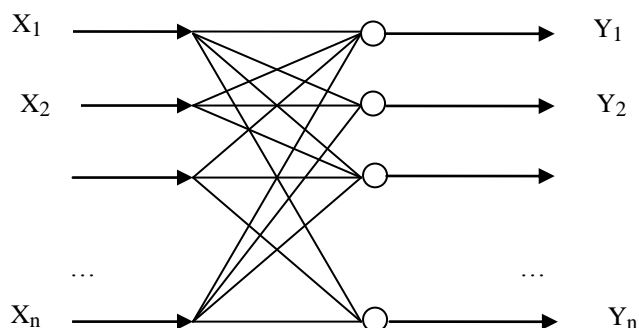


Рис. 2.11. Одношарова нейронна мережа

Такі нейронні мережі можна навчати за допомогою алгоритму навчання за Δ -правилом [18].

Алгоритм навчання з Δ -правилом:

1 крок: ініціалізація матриці ваг (і порогів, у разі використання порогової функції активації) випадковим чином.

2 крок: пред'явлення нейронної мережі образу (на вхід подаються значення з навчальної вибірки – вектор X), береться відповідний вихід (вектор D).

3 крок: обчислення вихідних значень нейронної мережі (вектор Y).

4 крок: обчислення для кожного нейрона величини розбіжності реального результату з очікуваним.

$$\varepsilon_i = (d_i - y_i),$$

де d_i – бажане вихідне значення на i -нейроні, y_i – реальне значення на i -нейроні.

5 крок: зміна ваг (і порогів при використанні порогової функції) за формулами:

$$W_{ij}(t+1) = W_{ij}(t) - \eta \cdot \varepsilon_i \cdot x_j,$$

$$\theta_i(t+1) = \theta_i(t) - \eta \cdot \varepsilon_i,$$

де t – номер поточної ітерації циклу навчання, W_{ij} – вага зв'язку j -входу з i -нейроном, η – коефіцієнт навчання, задається від 0 до 1, x_j – вхідна значення, θ_i – порогове значення i -нейрона.

6 крок: перевірка умови продовження навчання (обчислення значення помилки та/або перевірка заданої кількості ітерацій). Якщо навчання не завершено – то знову крок 2, інакше закінчуємо навчання.

Приклад 2.3.1.

Завдання. Прорахувати одну ітерацію циклу навчання за Δ -правилом одношарової бінарної неоднорідної нейронної мережі, що складається з 2 нейронів і має функції активації: гіперболічний тангенс ($k=1$) і порогову функцію ($T=0,7$). В якості навчальної вибірки використовувати таблицю істинності для операцій еквівалентності і диз'юнкції (не використовувати перший рядок таблиці). Синаптичні ваги задати випадковим чином.

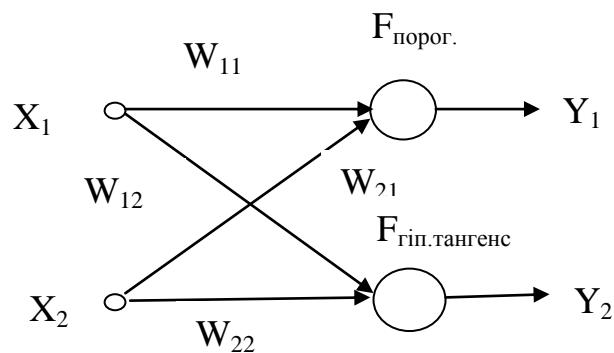
Опис процесу рішення. Для навчання нейронної мережі за Δ -правилом необхідно таке:

- 1) графічно відобразити структуру нейронної мережі. Визначити розмірність матриці синаптичних ваг;
- 2) визначити навчальну вибірку, представивши її в табличному вигляді;
- 3) вибрати вхідні дані, на яких буде розглядатися ітерація циклу навчання;

4) наступні алгоритми навчання за Δ -правилом – прорахувати одну ітерацію циклу і представити нові синаптичні ваги в матричному вигляді.

Рішення.

1) За завданням нейронна мережа складається з двох нейронів, значить, входів у одношарової нейронної мережі буде 2 і виходів 2, а синаптичних ваг 4. Перший нейрон має порогову функцію активації, другий – гіперболічний тангенс.



1. За завданням нейронна мережа бінарна, тому на її входи можуть подаватися лише нулі та одиниці, так як входів 2, то можливих комбінацій вхідних значень буде 4 (навчальна вибірка складатиметься з 4 векторів). Вихід першого нейрона згідно з завданням відповідає оператору еквівалентності, а другого – диз'юнкції. Тому таблиця з навчальною вибіркою буде виглядати таким чином:

2.

X_1	X_2	D_1	D_2
0	0	1	0
0	1	0	1
1	0	0	1
1	1	1	1

3. Нехай в якості вектора навчання буде розглядатися 3-й рядок таблиці.

4. Наступні алгоритми навчання за Δ -правилом – виконаємо 6 кроків

1 крок: задамо матрицю ваг випадковим чином з інтервалу $[0,1]$:

W_{ij}	1	2
1	0.7	0.9
2	0.5	0.2

2 крок: вектор $X = \{1, 0\}$, вектор $D = \{0, 1\}$.

3 крок: обчислення вихідних значень нейронної мережі (вектор Y).

$$T = 0.7;$$

$$S_1 = x_1 \cdot W_{11} + x_2 \cdot W_{21} = 1 \cdot 0.7 + 0 \cdot 0.5 = 0.7;$$

$$Y_1 = \begin{cases} 1, \text{при } S_1 \geq T \\ 0, \text{при } S_1 < t \end{cases} = \begin{cases} 1, \text{при } 0.7 \geq 0.7 \\ 0, \text{при } 0.7 < 0.7 \end{cases} = 1.$$

$$S_2 = x_1 \cdot W_{12} + x_2 \cdot W_{22} = 1 \cdot 0.9 + 0 \cdot 0.2 = 0.9,$$

$$Y_2 = th(S_2) = \frac{e^{0.9} - e^{-0.9}}{e^{0.9} + e^{-0.9}} \approx 0,716.$$

4 крок:

$$\varepsilon_1 = (d_1 - y_1) = (0 - 1) = -1,$$

$$\varepsilon_2 = (d_2 - y_2) = (1 - 0,716) = 0,284$$

5 крок: задаємо – коефіцієнт навчання від 0 до 1 і змінюємо ваги:

$$\eta = 0.8,$$

$$W_{11}(2) = W_{11}(1) - 0.8 \cdot \varepsilon_1 \cdot x_1 = 0.7 - 0.8 \cdot (-1) \cdot 1 = 1.5,$$

$$W_{21}(2) = W_{21}(1) - 0.8 \cdot \varepsilon_1 \cdot x_2 = 0.5 - 0.8 \cdot (-1) \cdot 0 = 0.5,$$

$$\theta_1(2) = \theta_1(t) - 0.8 \cdot \varepsilon_1 = 0.7 - 0.8 \cdot (-1) = 1.5,$$

$$W_{12}(2) = W_{12}(1) - 0.8 \cdot \varepsilon_2 \cdot x_1 = 0.9 - 0.8 \cdot (0.284) \cdot 1 = 0.673,$$

$$W_{22}(2) = W_{22}(1) - 0.8 \cdot \varepsilon_2 \cdot x_2 = 0.2 - 0.8 \cdot (0.284) \cdot 0 = 0.2.$$

$W_{ij}(1)$	1	2
1	1.5	0.673
2	0.5	0.2

бкрок: обчислимо середньоквадратичну помилку (можна вибрати інші методи оцінки помилки)

$$\varepsilon = \sum_{i=1}^N (d_i - y_i)^2 = \sum_{i=1}^N \varepsilon_i^2 = \varepsilon_1^2 + \varepsilon_2^2 = (-1)^2 + (0.284)^2 = 1.081,$$

N - кількість нейронів.

Вправи для самостійного розв'язання

1. Прорахувати одну ітерацію циклу навчання за Δ -правилом одношарової бінарної однорідної нейронної мережі, що складається з 2 нейронів і має порогову функцію активації ($T=0,7$). В якості навчальної вибірки використовувати таблицю істинності для операцій диз'юнкції та імплікації (не використовувати перший рядок таблиці). Синаптичні ваги задати випадковим чином.
2. Прорахувати одну ітерацію циклу навчання за Δ -правилом одношарової бінарної однорідної нейронної мережі, що складається з 2 нейронів і має лінійну функцію активації ($k=0,6$). В якості навчальної вибірки використовувати таблицю істинності для операцій кон'юнкції і диз'юнкції (не використовувати перший рядок таблиці). Синаптичні ваги задати випадковим чином.
3. Прорахувати одну ітерацію циклу навчання за Δ -правилом одношарової бінарної однорідної нейронної мережі, що складається з 2 нейронів і має сигмоїдальную функцію активації ($k=1$). В якості навчальної вибірки використовувати таблицю істинності для операцій імплікації і кон'юнкції (не використовувати перший рядок таблиці). Синаптичні ваги задати випадковим чином.
4. Прорахувати одну ітерацію циклу навчання за Δ -правилом одношарової бінарної однорідної нейронної мережі, що складається

з 2 нейронів і має функцію активації гіперболічний тангенс ($k=1$). В якості навчальної вибірки використовувати таблицю істинності для операцій еквівалентності та імплікації (не використовувати перший рядок таблиці). Синаптичні ваги задати випадковим чином.

5. Прорахувати одну ітерацію циклу навчання за Δ -правилом одношарової бінарної неоднорідної нейронної мережі, що складається з 2 нейронів і має функції активації: гіперболічний тангенс ($k=2$) і порогову функцію ($T=0,5$). В якості навчальної вибірки використовувати таблицю істинності для операцій еквівалентності і кон'юнкції (не використовувати перший рядок таблиці). Синаптичні ваги задати випадковим чином.

6. Прорахувати одну ітерацію циклу навчання за Δ -правилом одношарової бінарної неоднорідної нейронної мережі, що складається з 2 нейронів і має функції активації: сигмоїдальну ($k=1$) і лінійну ($k=0,6$). В якості навчальної вибірки використовувати таблицю істинності для операцій імплікації і кон'юнкції (не використовувати перший рядок таблиці). Синаптичні ваги задати випадковим чином.

7. Прорахувати одну ітерацію циклу навчання за Δ -правилом одношарової бінарної неоднорідної нейронної мережі, що складається з 2 нейронів і має функції активації: лінійну ($k=0,7$) і порогову ($T = 0,75$). В якості навчальної вибірки використовувати таблицю істинності для операцій кон'юнкції та еквівалентності (не використовувати перший рядок таблиці). Синаптичні ваги задати випадковим чином.

8. Прорахувати одну ітерацію циклу навчання за Δ -правилом одношарової бінарної неоднорідної нейронної мережі, що складається з 2 нейронів і має функції активації: порогову ($T=0,8$) і сигмоїдальну ($k=1$). В якості навчальної вибірки використовувати таблицю істинності для операцій кон'юнкції і імплікації (не використовувати перший рядок таблиці). Синаптичні ваги задати випадковим чином.

9. Прорахувати одну ітерацію циклу навчання за Δ -правилом одношарової бінарної неоднорідної нейронної мережі, що складається з 2 нейронів і має функції активації: гіперболічний тангенс ($k=2$) і лінійну ($k=0,8$). В якості навчальної вибірки використовувати таблицю істинності для операцій диз'юнкції і еквівалентності (не використовувати перший рядок таблиці). Синаптичні ваги задати випадковим чином.
10. Прорахувати одну ітерацію циклу навчання за Δ -правилом одношарової бінарної неоднорідної нейронної мережі, що складається з 2 нейронів і має функції активації: гіперболічний тангенс ($k=2$) і сигмоїдальную ($k=0,9$). В якості навчальної вибірки використовувати таблицю істинності для операцій імплікації і диз'юнкції (не використовувати перший рядок таблиці). Синаптичні ваги задати випадковим чином.
11. Прорахувати одну ітерацію циклу навчання за Δ -правилом одношарової бінарної однорідної нейронної мережі, що складається з 3 нейронів і має функцію активації гіперболічний тангенс ($k=3$). В якості навчальної вибірки використовувати таблицю істинності для $X1 \& X2 \rightarrow X3$, $X1 \& X2$ і $X2 \rightarrow X3$ (не використовувати перший рядок таблиці). Синаптичні ваги задати випадковим чином.
12. Прорахувати одну ітерацію циклу навчання за Δ -правилом одношарової бінарної однорідної нейронної мережі, що складається з 3 нейронів і має сигмоїдальную функцію активації ($k=1$). В якості навчальної вибірки використовувати таблицю істинності для $X1 \rightarrow X2 \& X3$, $X1 \& X2$ і $X1 \& X3$ (не використовувати перший рядок таблиці). Синаптичні ваги задати випадковим чином.
13. Прорахувати одну ітерацію циклу навчання за Δ -правилом одношарової бінарної однорідної нейронної мережі, що складається з 3 нейронів і має лінійну функцію активації ($k=0,9$). В якості навчальної вибірки використовувати таблицю істинності для $X3 \rightarrow X1 \& X2$, $X2 \& X3$, $X2 \rightarrow X3$ (не використовувати перший рядок таблиці). Синаптичні ваги задати випадковим чином.

14. Прорахувати одну ітерацію циклу навчання за Δ -правилом одношарової бінарної однорідної нейронної мережі, що складається з 3 нейронів і має порогову функцію активації ($T=0,4$). В якості навчальної вибірки використовувати таблицю істинності для $(X_2 \rightarrow X_1) \& X_3$ (не використовувати перший рядок таблиці). Синаптичні ваги задати випадковим чином.

15. Прорахувати одну ітерацію циклу навчання за Δ -правилом одношарової аналогової однорідної нейронної мережі, що складається з 3 нейронів і має лінійну функцію активації ($k=0,9$). Синаптичні ваги і навчальну вибірку задати випадковим чином (не нулі).

16. Прорахувати одну ітерацію циклу навчання за Δ -правилом одношарової аналогової однорідної нейронної мережі, що складається з 3 нейронів і має сигмоїдальную функцію активації ($k=0,8$). Синаптичні ваги і навчальну вибірку задати випадковим чином (не нулі).

17. Прорахувати одну ітерацію циклу навчання за Δ -правилом одношарової аналогової однорідної нейронної мережі, що складається з 3 нейронів і має порогову функцію активації ($T=0,8$). Синаптичні ваги і навчальну вибірку задати випадковим чином (не нулі).

18. Прорахувати одну ітерацію циклу навчання за Δ -правилом одношарової аналогової однорідної нейронної мережі, що складається з 3 нейронів і має функцію активації – гіперболічний тангенс ($k=1$). Синаптичні ваги і навчальну вибірку задати випадковим чином (не нулі).

19. Прорахувати одну ітерацію циклу навчання за Δ -правилом одношарової аналогової неоднорідної нейронної мережі, що складається з 3 нейронів і має функції активації: сигмоїдальную ($k=1$), лінійну ($k=0,8$) і порогову ($T=0,5$). Синаптичні ваги і навчальну вибірку задати випадковим чином (не нулі).

20. Прорахувати одну ітерацію циклу навчання за Δ -правилом одношарової аналогової неоднорідною нейронної мережі, що

складається з 3 нейронів і має функції активації: гіперболічний тангенс ($k=1$), сигмоїдальну ($k=0,8$) і порогову ($T=0,6$). Синаптичні ваги і навчальну вибірку задати випадковим чином (не нулі).

2.3.2.2. Алгоритм зворотного поширення помилки

Багатошарова штучна нейронна мережа (рис. 2.12) може містити довільну кількість шарів (K), кожен шар складається з декількох нейронів, число яких також може бути довільно (H_k – кількість нейронів у шарі), кількість входів n , кількість виходів $H = H_k$ – числу нейронів у вихідному (останньому) шарі.

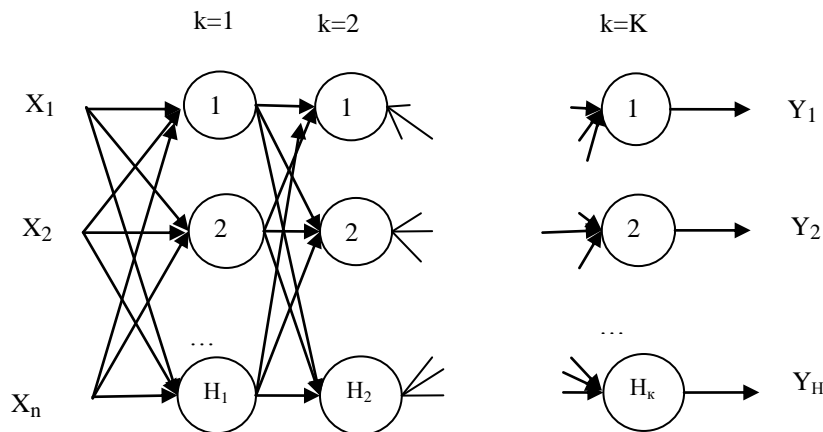


Рис. 2.12. Багатошарова нейронна мережа прямого розповсюдження

Шари між першим і останнім називаються проміжними або прихованими. Ваги в такій мережі мають три індекси i – номер нейрона наступного шару, для якого вхідний зв'язок, j – номер входу або нейрона поточного шару, для якого вихідний зв'язок, k – номер поточного шару нейронної мережі (для входів, вектора $X, k=0$). Багатошарові нейронні мережі прямого поширення навчаються методом зворотного поширення помилки.

Алгоритм навчання методом зворотного поширення помилки:

1 крок: ініціалізація матриць ваг випадковим чином (у циклах).

2 крок: пред'явлення нейронної мережі образу (на вхід подаються значення навчальної вибірки – вектор X) і береться відповідний вихід (вектор D).

3 крок (прямий прохід): обчислення в циклах виходів всіх шарів й отримання вихідних значень нейронної мережі (вектор Y).

$$Y_i^k = f \left(\sum_{j=0}^{H_{k-1}} W_{ij}^k \cdot Y_j^{k-1} \right),$$

$$Y_j^0 = x_j, \quad Y_0^{k-1} = 1, \quad x_0 = 1,$$

де Y_i^k – вихід i -нейрона k -шару, f – функція активації, W_{ij}^k – синаптична зв'язок між j -нейроном шару $k-1$ і i -нейроном шару k , x_j – вхідне значення.

4 крок (зворотний прохід): зміна ваг у циклах за формулами:

$$W_{ij}^k(t+1) = W_{ij}^k(t) - \eta \cdot \delta_i^k \cdot Y_j^{k-1},$$

$$\delta_i^k = (d_i - Y_i) \cdot Y_i \cdot (1 - Y_i),$$

для останнього (вихідного) шару,

$$\delta_i^k = Y_i \cdot (1 - Y_i) \cdot \sum_{i=1}^{H_{i+1}} \delta_i^{k+1} \cdot W_i^{k+1}$$

для проміжних шарів, де t -номер поточної ітерації циклу навчання (номер епохи), η – коефіцієнт навчання задається від 0 до 1, Y_i^k – вихід i -нейрона k -шару, W_{ij}^k – синаптичний зв'язок між j -нейроном шару $k-1$ і i -нейроном шару k , d_i – бажане вихідне значення на i -нейроні, Y_i – реальне значення на i -нейроні вихідного шару.

5 крок: перевірка умови продовження навчання (обчислення значення помилки та/або перевірка заданої кількості ітерацій). Якщо навчання не завершено, то крок 2, інакше закінчуємо навчання. Середньоквадратична помилка обчислюється таким чином:

$$\varepsilon = \frac{1}{Q} \cdot \sum_{q=1}^Q \sum_{i=1}^H (d_i - Y_i)^2,$$

де Q – загальна кількість прикладів, H – кількість нейронів у вихідному шарі.

Приклади розв'язання задач

Приклад 2.3.2.

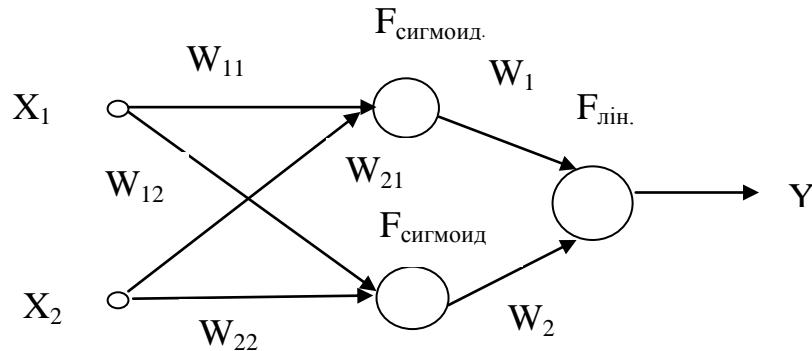
Прорахувати одну ітерацію циклу навчання методом зворотного поширення помилки багатошарової бінарної неоднорідної нейронної мережі, що складається з 2 шарів, причому в першому шарі знаходиться 2 нейрона і використовується сигмоїдальна функція активації ($k=0,9$), а в другому – 1, лінійна ($l=0,7$) функція. В якості навчальної вибірки використовувати таблицю істинності для операції «штрих Шеффера» (не використовувати перший рядок таблиці). Синаптичні ваги задати випадковим чином.

Опис процесу рішення. Для навчання нейронної мережі методом зворотного поширення помилки необхідно:

1. Графічно відобразити структуру нейронної мережі. Визначити розмір та кількість матриць синаптичних ваг (для кожного шару своя матриця).
2. Визначити навчальну вибірку, представивши її в табличному вигляді.
3. Вибрати вхідні дані, на яких буде розглядатися ітерація циклу навчання.
4. Наступні алгоритми навчання методом зворотного навчання помилки – прорахувати одну ітерацію циклу і представити нові синаптичні ваги в матричному вигляді.

Розв'язок

1. За завданням нейронна мережа складається з трьох нейронів, два вхідних, один вихідний, значить синаптичних ваг 6. Перший шар нейронів має сигмоїдальну функцію активації, другий – лінійну.



2. За завданням нейронна мережа бінарна, тому на її входи можуть подаватися лише нулі та одиниці, так як входів 2, то можливих комбінацій вхідних значень буде 4 (навчальна вибірка складатиметься з 4 векторів). Вихід нейронної мережі згідно з завданням відповідає оператору «штрих Шеффера». Тому таблиця з навчальною вибіркою буде виглядати таким чином:

X_1	X_2	D
0	0	1
0	1	1
1	0	1
1	1	0

3. Нехай в якості вектора навчання буде розглядатися 2-ий рядок таблиці.
 4. Дотримуючись алгоритму навчання за Δ -правилом, виконаємо 5 кроків:

1 крок: задамо матрицю ваг випадковим чином з інтервалу $[0, 1]$:

$W_{ij}(1)$	1	2
1	0.6	0.9
2	0.1	0.5
$W_g(1)$	1	2
	0.3	0.8

2 крок: вектор $X=\{0,1\}$, $D =\{1\}$.

3 крок (прямий прохід): обчислення в циклах виходів всіх шарів й отримання вихідних значень нейронної мережі (вектор Y).

$$S_1 = x_1 \cdot W_{11} + x_2 \cdot W_{21} = 0 \cdot 0.6 + 1 \cdot 0.1 = 0.1;$$

$$S_2 = x_1 \cdot W_{12} + x_2 \cdot W_{22} = 0 \cdot 0.9 + 1 \cdot 0.5 = 0.5$$

$$k = 0.9,$$

$$Y_1 = \frac{1}{1 + e^{-S_1 \cdot k}} = \frac{1}{1 + e^{-0.1 \cdot 0.9}} = 0.5224,$$

$$Y_2 = \frac{1}{1 + e^{-S_2 \cdot k}} = \frac{1}{1 + e^{-0.5 \cdot 0.9}} = 0.61,$$

$$S_3 = Y_1 \cdot W_1 + Y_2 \cdot W_2 = 0.5224 \cdot 0.3 + 0.61 \cdot 0.8 = 0.6472 :$$

$$l = 0.7,$$

$$Y = l \cdot S_3 = 0.4513.$$

4 крок (зворотний прохід): зміна ваг:

$$\eta = 0.7,$$

$$\delta^2 = (d - Y) \cdot Y \cdot (1 - Y) = (1 - 0.4513) \cdot 0.4513 \cdot (1 - 0.4513) = 0.3587,$$

$$W_1(2) = W_1(1) - \eta \cdot \delta^2 \cdot Y_1 = 0.3 - 0.7 \cdot 0.3587 \cdot 0.5224 = 0.167,$$

$$W_2(2) = W_2(1) - \eta \cdot \delta^2 \cdot Y_2 = 0.8 - 0.7 \cdot 0.3587 \cdot 0.61 = 0.647,$$

$$\delta_1^1 = Y_1 \cdot (1 - Y_1) \cdot \sum_{i=1}^{H_{k+1}} \delta_i^{k+1} \cdot W_i^{k+1} = Y_1 \cdot (1 - Y_1) \cdot \delta^2 \cdot W_1 =$$

$$= 0.5224 \cdot (1 - 0.5224) \cdot 0.3587 \cdot 0.3 = 0.0268,$$

$$\delta_2^1 = Y_2 \cdot (1 - Y_2) \cdot \sum_{i=1}^{H_{k+1}} \delta_i^{k+1} \cdot W_i^{k+1} = Y_2 \cdot (1 - Y_2) \cdot \delta^2 \cdot W_2 =$$

$$= 0.61 \cdot (1 - 0.61) \cdot 0.3587 \cdot 0.8 = 0.0682,$$

$$W_{11}(2) = W_{11}(1) - \eta \cdot \delta_1^1 \cdot x_1 = 0.6 - 0.7 \cdot 0.0268 \cdot 0 = 0.6,$$

$$W_{12}(2) = W_{12}(1) - \eta \cdot \delta_2^1 \cdot x_1 = 0.9 - 0.7 \cdot 0.0682 \cdot 0 = 0.9,$$

$$W_{21}(2) = W_{21}(1) - \eta \cdot \delta_1^1 \cdot x_2 = 0.1 - 0.7 \cdot 0.0268 \cdot 1 = 0.0812,$$

$$W_{22}(2) = W_{22}(1) - \eta \cdot \delta_2^1 \cdot x_2 = 0.5 - 0.7 \cdot 0.0682 \cdot 1 = 0.4523.$$

$W_{ii}(1)$	1	2
1	0.6	0.9
2	0.0812	0.452

$W_g(1)$	1	2
	0.431	0.953

5 крок:

$$\varepsilon = \sum_{i=1}^H (d_i - Y_i)^2 = (1 - 0.4513)^2 = 0.237.$$

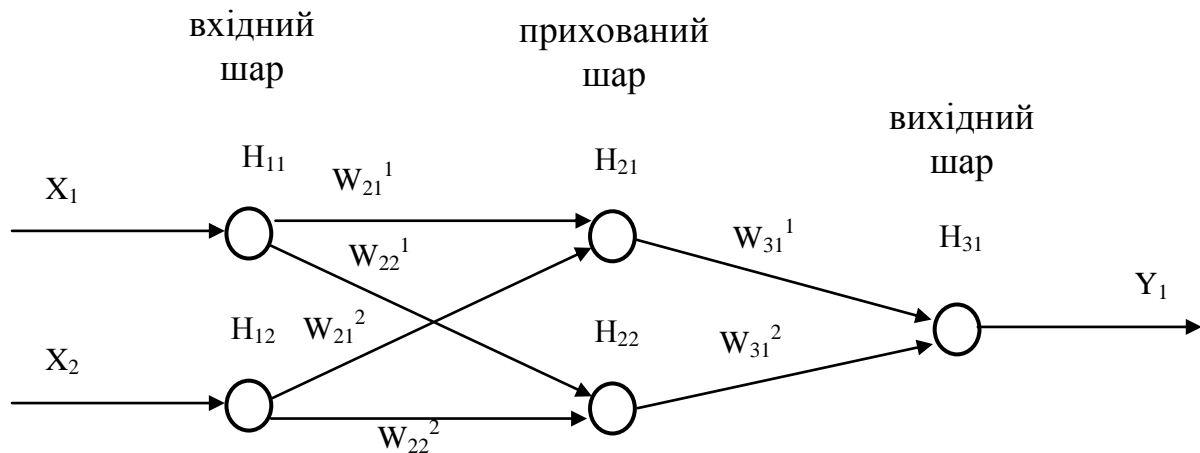
Так як ми розглядаємо одну ітерацію циклу навчання, у будь-якому випадку виходимо з циклу.

Приклад 2.3.3. Як приклад можна взяти навчальну вибірку:

x_1	x_2	y
1	0	2
2	1	6
4	2	16

тут x_1 і x_2 – вхідні параметри НМ, y – бажаний вихідний параметр.

Оскільки у функції, що апроксимується два вхідних параметра і один вихідний, то вибирається НМ з двома нейронами у вхідному шарі і одним у вихідному. Кількість нейронів прихованого шару приймемо рівним двом. Тобто формується мережа виду 2–2–1.



Як функції активації вибирається сігмоїдна функція з коефіцієнтом $\alpha = 1$. Початкові значення ваг синаптичних зв'язків приймаються рівними 0,5:

$$w_{21}^1 = w_{22}^1 = w_{21}^2 = w_{22}^2 = w_{31}^1 = w_{31}^2 = 0,5.$$

Оскільки вихідні значення x_1 , x_2 і y не лежать в межах $[0, 1]$, їх необхідно пронормувати, поділивши, наприклад, x_1 на 4, x_2 на 2, а y на 16. В результаті отримана нормована вибірка:

X_1	X_2	Y
0,25	0	0,125
0,5	0,5	0,375
1	1	1

Швидкість навчання приймається рівною $\eta = 0,2$.

Після підготовки можна приступати до навчання.

Крок 1. На входи НС подається перший вектор вхідних параметрів: $X_1 = 0,25$ і $X_2 = 0$. При цьому $Y_{\text{баж}} = 0,125$.

Виходи нейронів вхідного шару: $Y_{11} = 0,25$, $Y_{12} = 0$.

Для прихованого шару:

$$U_{21} = W_{21}^1 Y_{11} + W_{21}^2 Y_{12} = 0,5 \cdot 0,25 + 0,5 \cdot 0 = 0,125;$$

$$U_{22} = W_{22}^1 Y_{11} + W_{22}^2 Y_{12} = 0,5 \cdot 0,25 + 0,5 \cdot 0 = 0,125;$$

$$Y_{21} = 1/(1 + \exp(-\alpha U_{21})) = 1/(1 + \exp(-0,125)) = 0,5312;$$

$$Y_{22} = 1/(1 + \exp(-\alpha U_{22})) = 1/(1 + \exp(-0,125)) = 0,5312.$$

Для вихідного шару:

$$U_{31} = W_{31}^1 Y_{21} + W_{31}^2 Y_{22} = 0,5 \cdot 0,5312 + 0,5 \cdot 0,5312 = 0,5312;$$

$$Y_{31} = 1/(1 + \exp(-\alpha U_{31})) = 1/(1 + \exp(-0,5312)) = 0,6298.$$

Крок 2. Величина градієнта для вихідного нейрона:

$$EI_{31} = (Y_{31} - Y_{\text{жел}}) \cdot Y_{31} \cdot (1 - Y_{31}) = (0,6298 - 0,125) \cdot 0,6298 \cdot (1 - 0,6298) = 0,1177$$

Крок 3. Величини градієнтів для прихованого шару:

$$EI_{21} = Y_{21} \cdot (1 - Y_{21}) \cdot E_{31} \cdot W_{31}^1 = 0,5312 \cdot (1 - 0,5312) \cdot 0,1177 \cdot 0,5 = 0,01466;$$

$$EI_{22} = Y_{22} \cdot (1 - Y_{22}) \cdot E_{31} \cdot W_{31}^2 = 0,5312 \cdot (1 - 0,5312) \cdot 0,1177 \cdot 0,5 = 0,01466.$$

Крок 4. Корекція ваг синапсів:

$$W_{21}^1 = W_{21}^1 - \eta Y_{11} EI_{21} = 0,5 - 0,2 \cdot 0,25 \cdot 0,01466 = 0,4993;$$

$$W_{22}^1 = W_{22}^1 - \eta Y_{11} EI_{22} = 0,5 - 0,2 \cdot 0,25 \cdot 0,01466 = 0,4993;$$

$$W_{21}^2 = W_{21}^2 - \eta Y_{12} EI_{21} = 0,5 - 0,2 \cdot 0 \cdot 0,01466 = 0,5;$$

$$W_{22}^2 = W_{22}^2 - \eta Y_{12} EI_{22} = 0,5 - 0,2 \cdot 0 \cdot 0,01466 = 0,5;$$

$$W_{31}^1 = W_{31}^1 - \eta Y_{21} EI_{31} = 0,5 - 0,2 \cdot 0,5312 \cdot 0,1177 = 0,4875;$$

$$W_{31}^2 = W_{31}^2 - \eta Y_{22} EI_{31} = 0,5 - 0,2 \cdot 0,5312 \cdot 0,1177 = 0,4875.$$

Якщо при отриманих вагах на вхід НС подати той же вектор вхідних параметрів, то на виході буде $y = 0,6267$, що вже ближче до бажаного $Y_{\text{баж}} = 0,125$. Тобто даний цикл навчання наблизив відповідь НС до бажаного на величину $\Delta y = 0,6298 - 0,6267 = 0,0031$.

Оскільки навчальна вибірка не закінчилася, то кроки 1–4 повторюються аналогічно для наступного вектора вхідних параметрів.

Вправи для самостійного розв'язання

1. Прорахувати одну ітерацію циклу навчання методом зворотного поширення помилки багатошарової бінарної однорідної нейронної мережі, що складається з 2 шарів, причому в першому шарі знаходяться 2 нейрона, а у другому – 1. Функція активації нейронів мережі – порогова ($T = 0,6$) функція. В якості навчальної вибірки використовувати таблицю істинності для операції «виключає або» (не використовувати перший рядок таблиці). Синаптичні ваги задати випадковим чином.
2. Прорахувати одну ітерацію циклу навчання методом зворотного поширення помилки багатошарової бінарної однорідної нейронної мережі, що складається з 2 шарів, причому в першому шарі знаходяться 2 нейрона, а у другому – 1. Функція активації нейронів мережі – сигмоїдальна ($k=1$) функція. В якості навчальної вибірки використовувати таблицю істинності для операції імплікації (не використовувати перший рядок таблиці). Синаптичні ваги задати випадковим чином.
3. Прорахувати одну ітерацію циклу навчання методом зворотного поширення помилки багатошарової бінарної однорідної нейронної мережі, яка складається з 2 шарів, причому в першому шарі знаходяться 2 нейрона, а у другому – 1. Функція активації нейронів мережі – лінійна ($k = 0,6$) функція. В якості навчальної вибірки використовувати таблицю істинності для операції «штрих Шеффера» (не використовувати перший рядок таблиці). Синаптичні ваги задати випадковим чином.

4. Прорахувати одну ітерацію циклу навчання методом зворотного поширення помилки багат шарової бінарної однорідної нейронної мережі, що складається з 2 шарів, причому в першому шарі знаходяться 2 нейрона, а у другому – 1. Функція активації нейронів мережі – гіперболічний тангенс ($k = 1$). В якості навчальної вибірки використовувати таблицю істинності для операції «стрілка Пірса» (не використовувати перший рядок таблиці). Синаптичні ваги задати випадковим чином.

5. Прорахувати одну ітерацію циклу навчання методом зворотного поширення помилки багат шарової бінарної неоднорідної нейронної мережі, що складається з 2 шарів, причому в першому шарі знаходяться 2 нейрона і використовується сигмоїдальна функція активації ($k=0,9$), а в другому – 1, порогова ($T=0,7$). В якості навчальної вибірки використовувати таблицю істинності для операції «виключає або» (не використовувати перший рядок таблиці). Синаптичні ваги задати випадковим чином.

6. Прорахувати одну ітерацію циклу навчання методом зворотного поширення помилки багат шарової бінарної неоднорідної нейронної мережі, що складається з 2 шарів, причому в першому шарі знаходяться 2 нейрона і використовується лінійна функція активації ($k=0,5$), а у другому – 1, сигмоїдальна ($k=0,7$) функція. В якості навчальної вибірки використовувати таблицю істинності для операції імплікації (не використовувати перший рядок таблиці). Синаптичні ваги задати випадковим чином.

7. Прорахувати одну ітерацію циклу навчання методом зворотного поширення помилки багат шарової бінарної неоднорідної нейронної мережі, що складається з 2 шарів, причому в першому шарі знаходяться 2 нейрона і використовується порогова функція активації ($T=0,4$), а у другому – 1, лінійна ($k=0,6$) функція. В якості навчальної вибірки використовувати таблицю істинності для операції «штрих Шеффера» (не використовувати перший рядок таблиці). Синаптичні ваги задати випадковим чином.

8. Прорахувати одну ітерацію циклу навчання методом зворотного поширення помилки багатошарової бінарної неоднорідної нейронної мережі, що складається з 2 шарів, причому в першому шарі знаходяться 2 нейрона та використовується порогова функція активації ($T=0,6$), а у другому – 1, гіперболічний тангенс ($k=2$). В якості навчальної вибірки використовувати таблицю істинності для операції «стрілка Пірса» (не використовувати перший рядок таблиці). Синаптичні ваги задати випадковим чином.
9. Прорахувати одну ітерацію циклу навчання методом зворотного поширення помилки багатошарової аналогової неоднорідної нейронної мережі, що складається з 2 шарів, причому в першому шарі знаходяться 3 нейрона, а в другому – 2. Функція активації нейронів мережі – лінійна ($k=0,6$) функція. Синаптичні ваги і навчальну вибірку задати випадковим чином (не нулі).
10. Прорахувати одну ітерацію циклу навчання методом зворотного поширення помилки багатошарової аналогової неоднорідної нейронної мережі, що складається з 2 шарів, причому в першому шарі знаходяться 3 нейрона, а в другому – 2. Функція активації нейронів мережі – сигмоїдальна ($k=1$) функція. Синаптичні ваги і навчальну вибірку задати випадковим чином (не нулі).
11. Прорахувати одну ітерацію циклу навчання методом зворотного поширення помилки багатошарової аналогової неоднорідної нейронної мережі, що складається з 2 шарів, причому в першому шарі знаходяться 3 нейрона, а в другому – 2. Функція активації нейронів мережі – порогова ($T=0,65$) функція. Синаптичні ваги і навчальну вибірку задати випадковим чином (не нулі).
12. Прорахувати одну ітерацію циклу навчання методом зворотного поширення помилки багатошарової аналогової неоднорідної нейронної мережі, що складається з 2 шарів, причому в першому шарі знаходяться 3 нейрона, а в другому – 2. Функція активації нейронів мережі – гіперболічний тангенс ($k=3$) функція. Синаптичні ваги і навчальну вибірку задати випадковим чином (не нулі).
13. Прорахувати одну ітерацію циклу навчання методом зворот-

ного поширення помилки багатошарової аналогової неоднорідної нейронної мережі, що складається з 2 шарів, причому в першому шарі знаходяться 2 нейрона і використовується сигмоїдальна функція активації ($k=0,9$), у другому – 2, порогова ($T=0,7$). Синаптичні ваги і навчальну вибірку задати випадковим чином (не нулі).

14. Прорахувати одну ітерацію циклу навчання методом зворотного поширення помилки багатошарової аналогової неоднорідної нейронної мережі, що складається з 2 шарів, причому в першому шарі знаходяться 2 нейрона і використовується лінійна функція активації ($k=0,5$), у другому – 2, сигмоїдальна ($k=0,7$) функція. Синаптичні ваги і навчальну вибірку задати випадковим чином (не нулі).

15. Прорахувати одну ітерацію циклу навчання методом зворотного поширення помилки багатошарової аналогової неоднорідної нейронної мережі, що складається з 2 шарів, причому в першому шарі знаходяться 2 нейрона і використовується порогова функція активації ($T=0,4$), у другому – 2, лінійна ($k=0,6$) функція. Синаптичні ваги і навчальну вибірку задати випадковим чином (не нулі).

16. Прорахувати одну ітерацію циклу навчання методом зворотного поширення помилки багатошарової аналогової неоднорідної нейронної мережі, що складається з 2 шарів, причому в першому шарі знаходяться 2 нейрона і використовується порогова функція активації ($T=0,6$), у другому – 1, гіперболічний тангенс ($k=2$). Синаптичні ваги і навчальну вибірку задати випадковим чином (не нулі).

17. Прорахувати одну ітерацію циклу навчання методом зворотного поширення помилки багатошарової бінарної однорідної нейронної мережі, яка складається з 3 шарів, що використовує порогову функцію активації ($T=0,5$), в першому шарі 2 нейрона, у другому – 2, в третьому – 1. Синаптичні ваги і навчальну вибірку задати випадковим чином (не нулі).

18. Прорахувати одну ітерацію циклу навчання методом зворотного поширення помилки багатошарової бінарної однорідної нейронної мережі, яка складається з 2 шарів, що використовує порогову функцію активації ($T=0,5$), у першому шарі 3 нейрона, у другому – 1. В якості навчальної вибірки використовувати таблицю істинності для $X1 \rightarrow X2 \& X3$ (не використовувати перший рядок таблиці). Синаптичні ваги задати випадковим чином.

19. Прорахувати одну ітерацію циклу навчання методом зворотного поширення помилки багатошарової бінарної однорідної нейронної мережі, що складається з 2 шарів, використовує сигмоїдальную функцію активації ($k=0,5$), у першому шарі 3 нейрона, у другому – 1. В якості навчальної вибірки використовувати таблицю істинності для $(X1 \rightarrow X2) \& X3$ (не використовувати перший рядок таблиці). Синаптичні ваги задати випадковим чином.

20. Прорахувати одну ітерацію циклу навчання методом зворотного поширення помилки багатошарової аналогової неоднорідної нейронної мережі, що складається з 3 шарів, причому в першому шарі знаходяться 2 нейрона і використовується порогова функція активації ($T=0,6$), у другому – 2, гіперболічний тангенс ($k=2$), в третьому 1, лінійна ($k=0,7$). Синаптичні ваги і навчальну вибірку задати випадковим чином (не нулі).

Практичні рекомендації при використанні алгоритму ЗПП

1. **Вибір початкових значень.** Цей етап має вплив на досягнення мережею глобального мінімуму функції помилки і на швидкість сходження до мінімуму. З одного боку, значення початкових ваг не повинні бути дуже великими, інакше початкові входні сигнали в кожен прихований або входний клітинку потраплять в діапазон, де похідна сигмоїдної функції активації має дуже малу величину. З іншого боку, якщо початкові значення взяти достатньо малими, то мережний вхід у прихований або вихідний нейрон буде ближче до нуля, що призведе до дуже повільного навчання.

Загальне правило ініціювання початкових ваг полягає у виборі їх значень з рівномірно розподілених величин в інтервалі $(-0,5...0,5)$ – іноді, цей інтервал може бути трохи менше або більше, але не перевищувати ± 1 . Значення ваг можуть позитивними або негативними, оскільки остаточні ваги після навчання також можуть мати будь-який знак.

2. Тривалість навчання мережі. Пропонується використовувати дві серії даних під час навчання: серію навчальних образів і серію контрольних образів. Ці дві серії є окремими. Регулювання ваги засноване на навчальних образах, проте під час навчання помилка обчислюється з використанням контрольних образів. До тих пір, поки помилка для контрольних образів зменшується, процес навчання триває. При зростанні цієї помилки мережа починає втрачати свою здатність до узагальнення, і в цей момент навчання припиняється.

3. Кількість необхідних пар для навчання. Для співвідношення між числом образів P , що навчаються, кількістю регульованих ваг W і точністю класифікації ε запропоновано використовувати такий вираз: $w/P = \varepsilon$. Наприклад, для $\varepsilon = 0,1$ багатошарова мережа з 80 регульованими вагами зажадає 800 учнів образів, щоб бути впевненим у правильній класифікації 90 % пропонованих контрольних образів.

4. Подання даних. Для нейронної мережі легше навчитися набору різних станів, ніж відгуку з безперервним значенням. У багатьох завданнях вхідні і вихідні вектори мають складові в одному й тому ж діапазоні величин. Внаслідок того, що один з членів у виразі для коригування ваг є активацією комірки попереднього шару, нейрони, що мають нульову активацію, навчатися не будуть. Навчання може бути покращено в тому випадку, якщо вхідний вектор представлений у біполярній формі, а в якості функції активації використовується біполярна сигмоїда (біполярна сигмоїда дуже близька до функції гіперболічного тангенса).

5. Введення інерційної поправки. Показано, що пошук мінімуму функції помилок методом градієнтного спуску виявляється досить повільним, якщо швидкість навчання η мала, і призводить до значних осциляцій при великій швидкості η .

6. Модифікація функції активації. Діапазон функції активації повинен відповідати діапазону значень конкретного завдання. Бінарна сигмоїдна функція виду $f(x) = [1 + \exp(-x)]^{-1}$ з похідною

$$f'(x) = f(x) \cdot [1 - f(x)]$$

може бути змінена для перекриття будь-якого необхідного діапазону з центром при будь-якому значенні x і необхідному нахилі. Сигмоїда може мати розширений діапазон, щоб відображати значення в інтервалі $[a, b]$ для будь-яких a і b . Для цього потрібно ввести параметри

$$\gamma = b - a; \quad \rho = -a.$$

Тоді сигмоїдна функція:

$$g(x) = \gamma f(x) - \rho$$

має необхідні властивості, тобто діапазон $[a, b]$. Її похідна виражається як:

$$g'(x) = \gamma^{-1} [\rho + g(x)] \cdot [\gamma - \rho - g(x)]$$

Нахил сигмоїдної функції може бути змінений за допомогою введеного параметра γ .

7. Кількість нейронів у прихованому шарі. Підхід при виборі прихованих нейронів полягає в тому, що на першому етапі кількість таких нейронів береться завідомо більшою ніж потрібно, а далі, по мірі навчання мережі, зайві нейрони прибираються. Всі нейрони, які не вносять вклад у рішення або дають інформацію, що не вимагається у наступному шарі, розглядаються як зайві і видаляються з прихованого шару. На практиці вважається, що мережа досягла збіжності, якщо різниця між необхідним і дійс-

ним виходами не перевищує 0,1. Якщо два прихованих нейрони дають приблизно однаковий вихід для всіх навчальних прикладів, то тільки один з них дійсно необхідний, так як обидва нейрони переносять однакову інформацію. Після видалення клітинок, які не дають внеску в рішення, значення ваги зменшення мережі повинні бути змінені шляхом перенавчання мережі для отримання необхідних характеристик.

Класичний метод ЗПП відноситься до алгоритмів з лінійною збіжністю. Його відомими недоліками є: невисока швидкість збіжності (велика кількість ітерацій), можливість сходиться не до глобального, а до локальних рішень. Можливий також параліч мережі – більшість нейронів функціонують при дуже великих значеннях аргументу функції активації, тобто на пологій ділянці (т. я. помилка пропорційна похідній, яка на даних ділянках мала, то процес навчання практично завмирає). Для усунення цих недоліків були запропоновані численні модифікації алгоритму ЗПП.

Навчання без вчителя. Головна риса, що робить навчання без вчителя привабливим, – це його «самостійність». Процес навчання як і у випадку з вчителем, полягає в підстроюванні ваг синапсів. Очевидно, що підстроювання ваг синапсів може проводитися тільки на підставі інформації, доступної в нейроні, тобто інформації про його стан, вже наявних вагових коефіцієнтів і поданому векторі X . Виходячи з цього і за аналогією з відомими принципами самоорганізації нервових клітин, побудовані алгоритми навчання Хеба і Кохонена. Загальна ідея даних алгоритмів полягає в тому, що в процесі самонавчання шляхом відповідної корекції вагових коефіцієнтів посилюються зв'язки між збудженими нейронами.

2.3.3. Типи нейронних мереж

Багатошарові нейронні мережі – це мережі з сигмоїдними передавальними функціями є найбільш загальними, універсальними мережними архітектурами. Є різні структури багатошаро-

вих мереж: з послідовними, перехресними і зворотними зв'язками, з фіксованою та змінною структурою. З ростом числа шарів зростають як складність побудови мережі, так і якість її роботи. Багатошарові нейронні мережі є універсальними апроксиматорами.

Мережа Кохонена – кластерний аналіз, розпізнавання образів, класифікація. Шар Кохонена самонавчається.

Нейронні мережі зустрічного поширення – розпізнавання та відновлення образів, стиснення даних (з утратою інформації), статистичний аналіз. Мережа містить два шари з послідовними зв'язками: перший шар Кохонена, другий Гросберга. **Переваги:** 1) мережа проста; 2) мережа швидко навчається; мережа дає можливість будувати функцію і зворотну до неї функцію, що знаходить застосування при вирішенні практичних завдань. **Недоліки.** Слабка теоретична проробка. Мережа не дає можливості будувати точні апроксимації.

Нейронні мережі Хопфілда і Хеммінга дозволяють просто та ефективно вирішити завдання відтворення образів за неповної та спотвореної інформації.

Мережі з радіальними базисними функціями (RBF) – двошарові мережі без зворотних зв'язків, яка містять прихований шар радіально-симетричних прихованих нейронів (шаблонний шар). Мережі RBF моделюють довільну нелінійну функцію за допомогою всього одного проміжного шару. Параметри лінійної комбінації у вихідному шарі можна повністю оптимізувати за допомогою добре відомих методів лінійної оптимізації, які працюють швидко і не відчують труднощів з локальними мінімумами, що заважають при навчанні з використанням алгоритму ЗПП. Тому мережа RBF навчається на порядок швидше, ніж з використанням алгоритму ЗПП. **Недоліками** мереж RBF є: мережі мають погані екстраполюючі властивості і виходять досить громіздкими при великій розмірності вектора входів. Модифікацією мережі RBF є радіальна базисна мережа з нульовою помилкою. Іншими модифікаціями є ймовірнісні нейронні мережі (PNN) та узагальнено-регресійна нейронна мережа (GRNN).

Імовірнісна нейронна мережа (PNN). Призначені для розв'язання імовірнісних задач і, зокрема, задач класифікації. Архітектура мережі PNN базується на архітектурі базисної мережі, але в якості другого шару використовують так званий конкуруючий шар, який підраховує ймовірність приналежності вхідного вектора до того чи іншого класу, і, в кінцевому рахунку, зіставляє вектор з тим класом, ймовірність приналежності до якого вище. Вихідне значення має ймовірнісний сенс. Мережа швидко навчається. **Недоліком** таких мереж є їх обсяг – мережі вимагають багато пам'яті і можуть повільно працювати.

Узагальнено-регресійна нейронна мережа (GRNN). Дана мережа аналогічна імовірнісній нейронній мережі, але вона призначена для вирішення завдань регресії. Як і мережа RBF, мережа GRNN не має здатності екстраполювати дані.

Лінійні нейронні мережі. Згідно із загальноприйнятим у науці принципом, якщо більш складна модель не дає кращих результатів, ніж більш проста, то з них слід віддати перевагу другій. У термінах апроксимації відображень найпростішою моделлю буде лінійна, у якій апроксимувальна (підгоняльна) функція визначається гіперплощиною. У задачі класифікації гіперплощина розміщується таким чином, щоб вона розділяла собою два класи (лінійна дискримінантна функція); у задачі регресії гіперплощина повинна проходити через задані точки. Лінійна модель задається рівнянням:

$$Y = X \cdot W + B,$$

де W – матриця ваг мережі, B – вектор зміщень.

На мові нейронних мереж лінійна модель представляється мережею без проміжних шарів, яка у вихідному шарі містить тільки лінійні елементи, тобто елементи з лінійною функцією активації. Ваги відповідають елементам матриці, а пороги – компонентам вектора зміщення.

2.4. Пакет Neural Networks Toolbox

Пакет Neural Networks Toolbox (нейронні мережі середовище MatLab) містить засоби для проектування, моделювання, навчання та використання множин відомих парадигм апарата штучних нейронних мереж від базових моделей перцептрона до сучасних асоціативних і самоорганізуючих мереж [12,13]. Пакет може бути використаний для вирішення багатьох різноманітних завдань, таких як обробка сигналів, нелінійне управління, фінансове моделювання і т.ін.

Для кожного типу архітектури та навчального алгоритму ШНМ є функції ініціалізації, навчання, адаптації, створення і моделювання, демонстрації та приклади застосування.

Приклади створення та використання нейронних мереж

1. **Функції створення нейронних мереж.** Функції даної групи дозволяють створити НС заданої структури (ненавчену).

Синтаксис: **Ім'я мережі = функція** (параметри мережі)

Наприклад, запис вигляду

net = newlind (p,t),

Опис: **newlind** – функція проектування лінійної НМ. Дана функція по матрицях вхідних і вихідних векторів методом найменших квадратів визначає ваги і зміщення лінійної НМ.

net = newgrnn (p,t,spread) – функція створення узагальнено-регресійної мережі;

net = newrbe (p,t,spread) – функція створення мережі з радіальними базисними елементами з нульовою помилкою на навчальній вибірці;

net = newpnn (p,t,spread) – функція створення ймовірнісної НМ;

net = newp (pr,s,tf,lf)– функція створення перцептрона (pr–матриця мінімальних і максимальних значень вхідних елементів,

s – число нейронів, tf – функція активації, за замовчуванням порогова, lf – функція, що реалізує алгоритм навчання пресептрона.

Функції моделювання НС

1. **Sim** – функція, що моделює роботу НС дозволяє розрахувати виходи навченої НС при заданих векторах входів.

2. **Gensim** – функція формує S-модель нейронної мережі для її запуску в середовищі Simulink.

Синтаксис: **Gensim (net,st)**, де *net* – ім'я створеної мережі, *st* – інтервал дискретизації (якщо НМ не має затримок, асоційованих з її входами або шарами, значення встановлюється таким, що дорівнює -1). Функція генерує нейромережний блок Simulink для подальшого моделювання НМ засобами цього пакету.

Приклад 2.4.1. Створити узагальнено-регресійну НМ (типу GRNN) з ім'ям, що реалізує функціональну залежність між входом і виходом виду $y = x^2$. Синтаксис: `net = newgrnn(p,t,spread)`, де *p* – матриця вхідних векторів, *t* – матриця цільових векторів, *spread* – відхилення (за замовчуванням 1,0). Використовувати такі експериментальні дані:

$$x = [-1 \ -0.8 \ -0.5 \ -0.2 \ 0 \ 0.1 \ 0.3 \ 0.6 \ 0.9 \ 1];$$

$$y = [1 \ 0.64 \ 0.25 \ 0.04 \ 0 \ 0.01 \ 0.09 \ 0.36 \ 0.81 \ 1].$$

Перевірку якості відновлення наведеної залежності здійснити, використовуючи дані контрольної вибірки $x_1 = [-0.9 \ -0.7 \ -0.3 \ 0.4 \ 0.8]$.

Процедура створення і використання даної НС описується таким чином:

```
>> x=[-1 -0.8 -0.5 -0.2 0 0.1 0.3 0.6 0.9 1]; завдання вхідних значень
```

```
>> y=[1 0.64 0.25 0.04 0 0.01 0.09 0.36 0.81 1]; завдання цільових значень
```

```
>>a=newgrnn(x,y,0.01); створення НС з відхиленням 0,01
```

```
>>y1=sim(a,[-0.9 -0.7 -0.3 0.4 0.8]) опитування мережі
```

```
y1 =
0.8200 0.5049 0.0316 0.0710 0.6390
```

Як видно, точність апроксимації в даному разі вийшла не дуже високою. Спробуємо використовувати мережу з радіальними базисними елементами типу `newrbe`:

```
>>a=newrbe(x,y);
>>y1=sim(a,[-0.9 -0.7 -0.3 0.4 0.8])
y1 =
0.8100 0.4900 0.0900 0.1600 0.6400
```

Неважко бачити, що застосування мережі даного типу призводить до точного відновлення заданої залежності.

Приклад 2.4.2. Розглянемо завдання відновлення деякої, взагалі кажучи, невідомої залежності за наявними експериментальними даними з використанням лінійної НС. Нехай експериментальна інформація задана значеннями

```
x = [1.0 1.5 3.0 -1.2],
y = [0.5 1.1 3.0 -1.0]
```

Створимо вектори входу і цілей:

```
>> x=[1.0 1.5 3.0 -1.2];
>> y=[0.5 1.1 3.0 -1.0];
```

Створимо лінійну нейронну мережу:

```
>>b=newlind(x,y); Створення НМ з ім'ям b.
```

Проведемо опитування мережі для значення входу, рівного 3,0 (цьому, згідно з експериментальними, відповідає цільове значення 3,0):

```
>>y1=sim(b, 3.0) Опитування мережі
y1 =
2.7003
```

Похибка відновлення за даними навчальної вибірки в даному випадку – 10%.

2.5. Використання Simulink при побудові НС

Пакет Neural Networks містить ряд блоків, які можуть бути безпосередньо використані для побудови НМ в середовищі Simulink, або застосовуватися разом з розглянутою функцією gensim. Для виклику набору блоків у командному рядку потрібно набрати команду *neural*, після виконання якої з'являється вікно (рис.2.13).

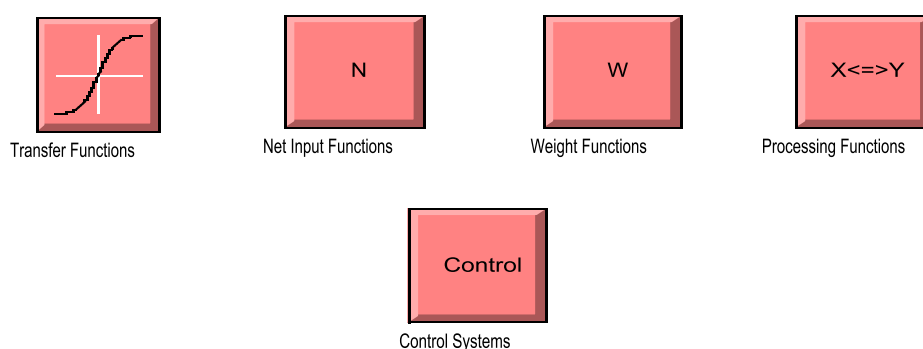


Рис.2.13. Основні неймережні блоки Simulink

Кожен з представлених на рис.2.13 блоків, у свою чергу, є набором деяких блоків. Блоки функції активації. Подвійне клацання на блоці призводить до появи бібліотеки блоків (рис.2.14). Кожен з цих блоків даної бібліотеки перетворює вектор, що подається на нього, на відповідний вектор тієї ж розмірності.

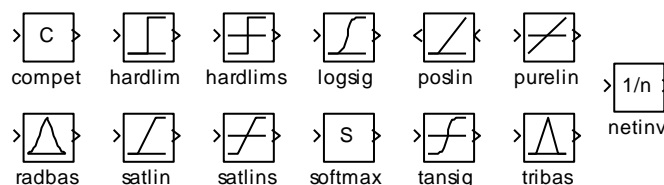


Рис.2.14. Бібліотека блоків функцій активації

Блоки перетворення входів мережі. Функції даної групи реалізують функції накопичення потенціалу нейрона у вигляді

поелементного добутку або сум зважених входів нейрона, а також обчислюють похідні від таких добутків або сум (рис.2.15).

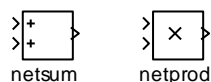


Рис.2.15. Вікно бібліотеки блоків перетворення сигналів

Блоки вагових коефіцієнтів (рис.2.16). Функції цієї групи виконують такі операції: зважування та обчислення відстаней у мережах з топологією (`dotprod` – функція надання входів P деяких ваг W . Повертає матрицю $Z = W \cdot P$; `normprod` – функція обчислення нормованого скалярного добутку (кожен елемент додатково ділиться на суму елементів відповідного стовпця-співмножника); `dist` – функція обчислення евклідової відстані; `negdist` – негативне евклідова відстань. Вектори в Simulink необхідно представляти як стовпці.

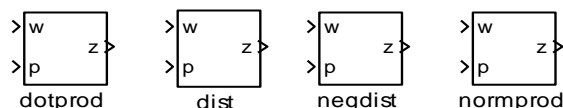


Рис.2.16. Бібліотека блоків вагових коефіцієнтів

Блоки нейромережних регуляторів. На рис.2.17 показаний набір блоків, об'єднаних у бібліотеку системи контролю. Дані блоки реалізують нейромережні регулятори трьох різних структур – регулятор з передбаченням, регулятор, заснований на використанні моделі нелінійної авторегресії з ковзним середнім (NARMA 1,2 Controller) і регулятор на основі еталонної моделі, які зручні при побудові та дослідженні моделей систем автоматичного керування, а також блок перегляду результатів.

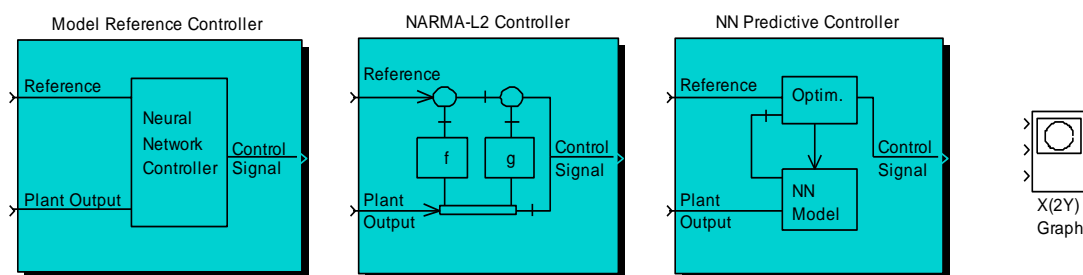


Рис.2.17. Вікно бібліотеки нейромережних регуляторів

Формування нейромережних моделей. Основною функцією для формування нейромережних моделей Simulink є функція `gensim`, що записана у формі

`Gensim (net,st)`, де `net` – ім'я створеної мережі, `st` – інтервал дискретизації (якщо НМ не має затримок, асоційованих з її входами або шарами, значення встановлюється такими, що дорівнюють -1). Функція генерує нейромережний блок Simulink для подальшого моделювання НМ засобами цього пакету.

Приклад 2.5.1. Створити двошарову НС прямої передачі сигналу (функція створення багатошарової НС записується як `newff`) з вектором мінімальних і максимальних значень входів $[0,1]$, п'ятьма нейронами в першому (прихованому) шарі і одним нейроном – у вихідному шарі, а потім сформуванати S-модель такої НМ. Дана задача вирішується таким чином:

```
>>net=newff([0 1],[5 1]); % Створення нової НМ
>>gensim(net)
```

Результат виконання функції `gensim` відображений на рис. 2.18.

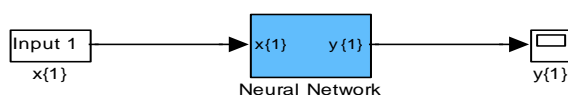


Рис. 2.18. Результат виконання функції `gensim`

Двічі клацаючи на блоці Neural Network, а потім на блоках Layer1 і Layer2, можна отримати детальну інформацію про структуру мережі (рис.2.19, 2.20)

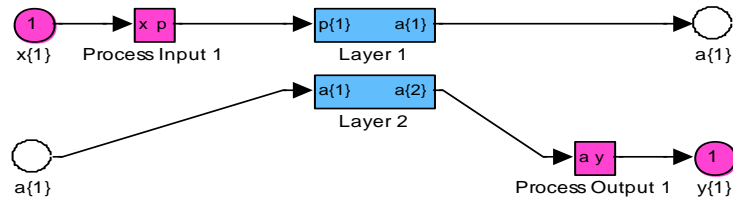
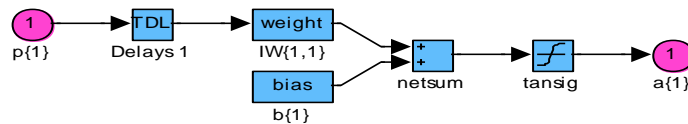
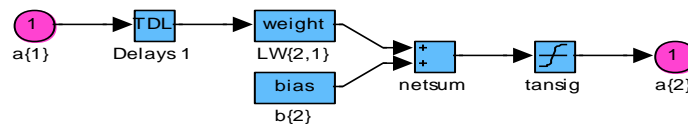


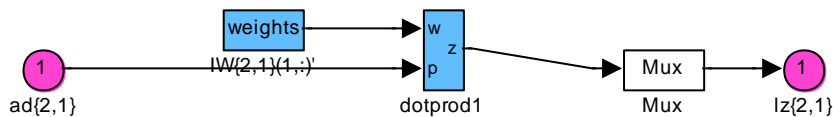
Рис. 2.19. Структура створеної НС



Структура 1-го шару



Структура 2-го шару



Структура вихідного шару

Рис.2.20. Структура створеної НС з детальною графічною інформацією

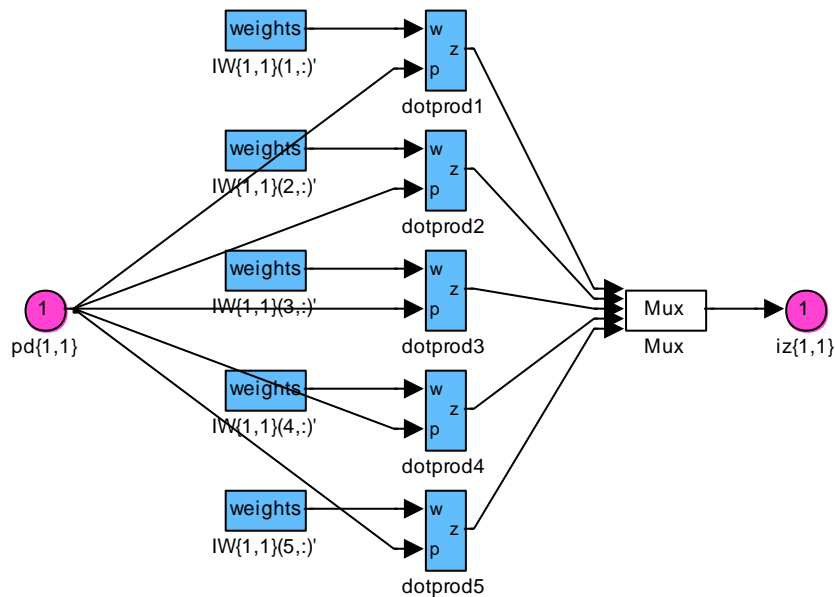


Рис. 2.21. Структура прихованого шару

Приклад 2.5.2. Нехай вхідний і цільовий вектори мають вигляд

$$p=[1 \ 2 \ 3 \ 4 \ 5];$$

$$t=[1 \ 3 \ 5 \ 7 \ 9];$$

Створимо лівів НС і протестуємо її за даними навчальної вибірки:

```
>> p=[1 2 3 4 5];
```

```
>> t=[1 3 5 7 9];
```

```
>> net=newlind(p,t); створення НМ
```

```
>> y=sim(net,p) опитування мережі
```

```
y =
```

```
1.0000 3.0000 5.0000 7.0000 9.0000
```

Запустимо Simulink командою

```
>>gensim(net,-1)
```

Це призведе до відкриття блок-діаграми, наведеної на рис. 2.18. У цьому випадку блок Input1 є стандартним блоком завдання константи (Constant). Змінимо значення за замовчуван-

ням на 4 і натиснемо кнопку ОК, а потім кнопку «Пуск» на панелі інструментів. Для виведення нового значення необхідно двічі клацнути на правій піктограмі блоку $u(1)$, воно буде дорівнює 7. Зі створеними мережами можна проводити різні експерименти, можливі в середовищі Simulink, наприклад, можливо вбудовування нейромережного регулятора в систему управління і моделювання останньої.

2.6. Система автоматичного управління з нейромережним регулятором на основі еталонної моделі

В якості прикладу моделювання в середовищі Simulink систем управління з використанням нейромережних регуляторів використовуємо приклад, що ілюструє систему управління з еталонною моделлю реактора повного перемішування (рис.2.22).

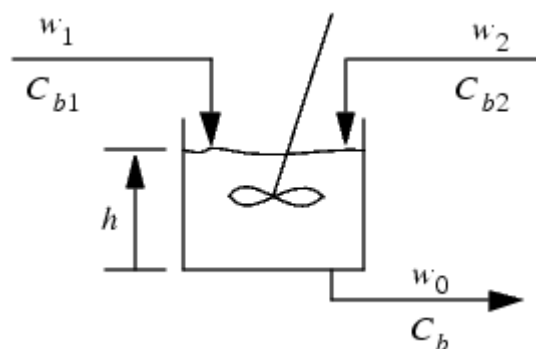


Рис.2.22. Реактор повного перемішування (w_1, w_2, w_0 – витрати, h – рівень, C_{b1}, C_{b2}, C_b – концентрації компонентів на вході і виході з реактора, відповідно)

Зміни концентрації в реакторі і на виході з нього, а також рівень рідини в реакторі (динамічна модель системи) представлені такою системою диференціальних рівнянь (отримані з рівнянь матеріального балансу):

$$\frac{dC_b(t)}{dt} = (C_{b1} - C_b(t)) \frac{w_1(t)}{h(t)} + (C_{b2} - C_b(t)) \frac{w_2}{h(t)} - \frac{k_1 C_b(t)}{(1 + k_2 C_b(t))^2},$$

$$\frac{dh(t)}{dt} = w_1(t) + w_2(t) - 0,2\sqrt{h(t)},$$

де $k_1 = 1$ и $k_2 = 1$ – константи, $C_{b1} = 24.9$, $C_{b2} = 0.1$.

Мета автоматичного управління - стабілізувати концентрацію на виході, регулюючи потік $w_1(t)$ (витрата $w_2(t) = 0.1$ є величиною постійною). Рівень резервуара $h(t)$ не є керованим параметром.

Відповідна динамічна модель об'єкта регулювання наведена на рис.2.23.

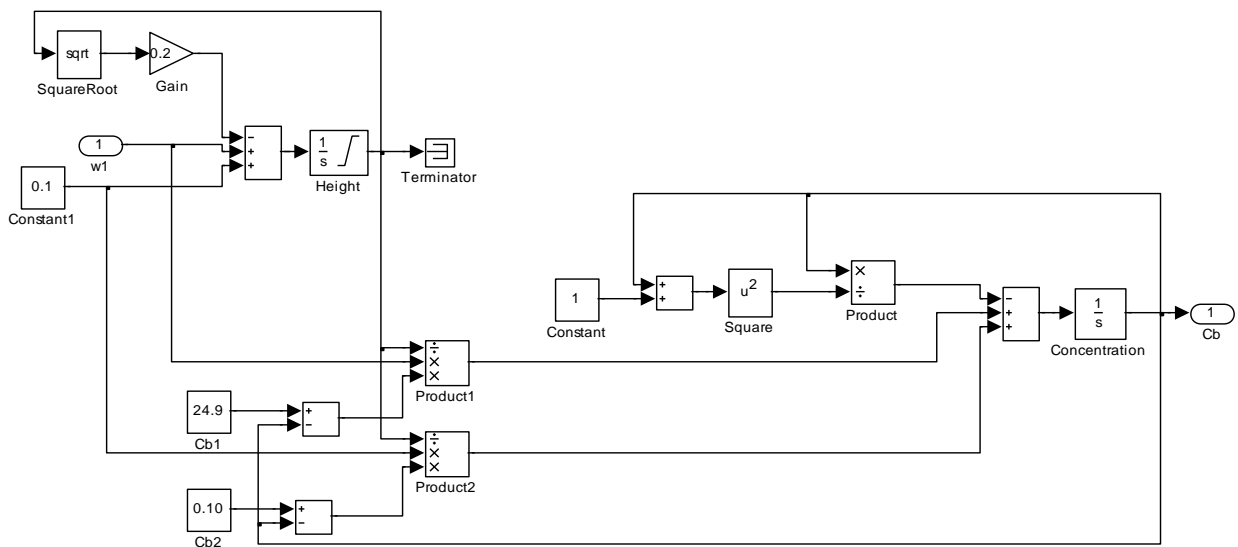


Рис.2.23. Структурна динамічна модель реактора

Структурна схема, що пояснює принцип побудови системи управління показана на рис. 2.24.

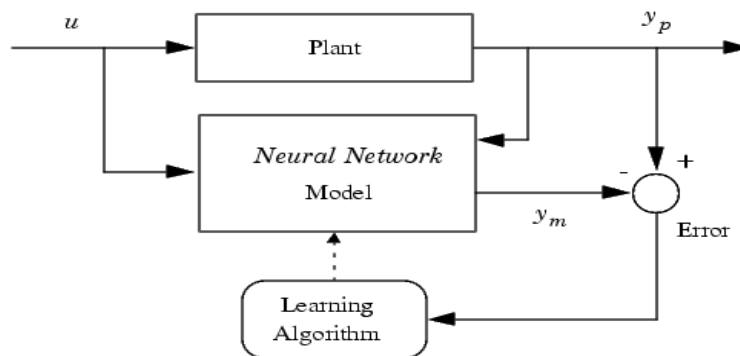


Рис.2.24. Система управління (Plant – об'єкт управління, Learning Algorithm – навчальний алгоритм)

Автоматичний регулятор складається з нейронної мережної моделі об'єкта і блоку оптимізації (рис.2.25). Блок оптимізації визначає оптимальне значення керуючого параметра (u), який подається на вхід об'єкта.

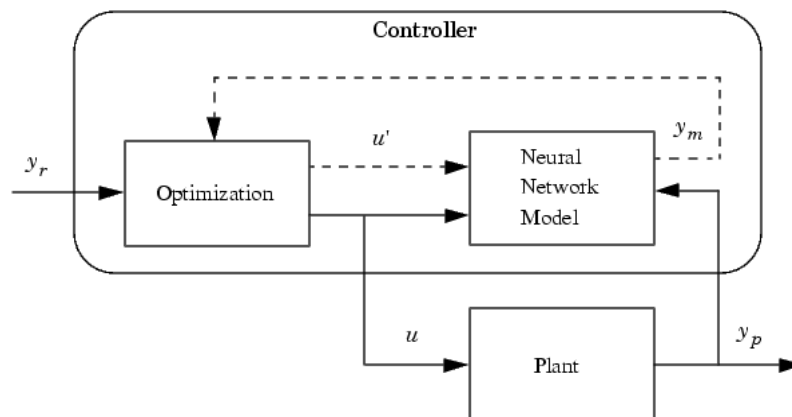


Рис.2.25. Нейронна мережа об'єкта і блок оптимізації

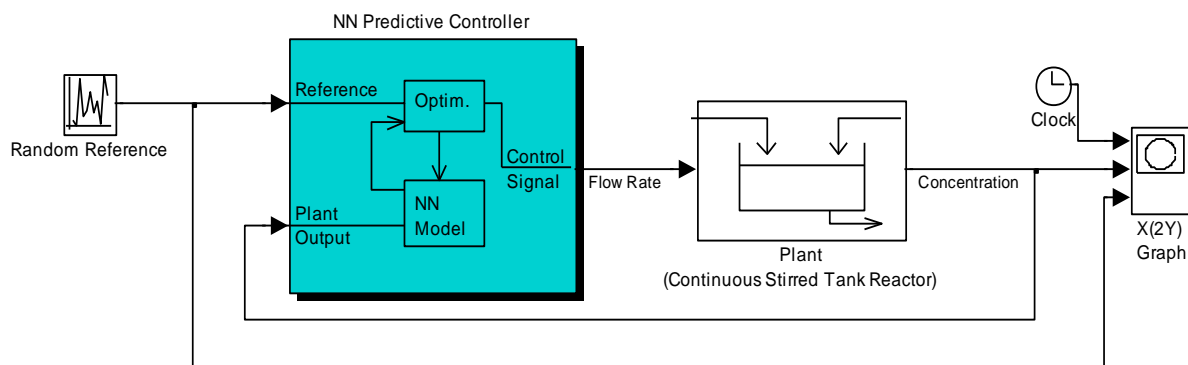
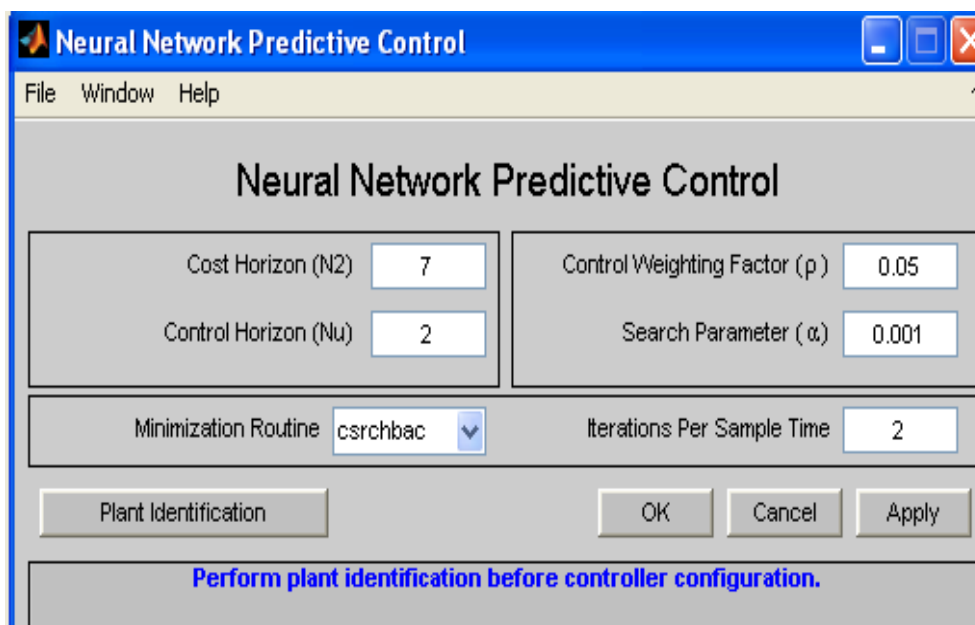


Рис.2.26. Структура нейромережної системи управління реактора повного перемішування (random reference – програма, що генерує дані, які навчаються, застосовуючи випадкові сигнали до Simulink моделі об'єкта)

На рис.2.27 показано, як будується НМ моделі регулятора у вікні Neural Network Predictive Control і вікні Plant Identification (ρ – ваговий фактор, α – параметр оптимізації). На рис. 2.28 показаний графік регулювання концентрації продукту на виході реактора.



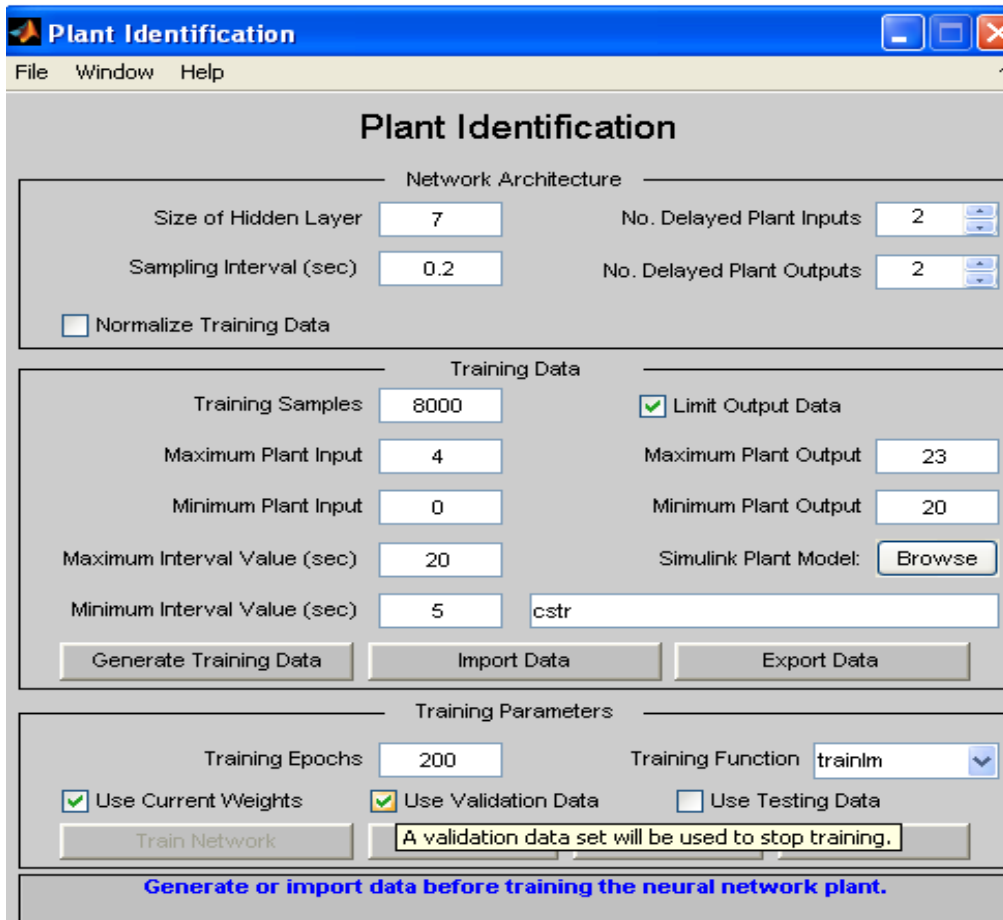


Рис.2.27. НМ моделі регулятора

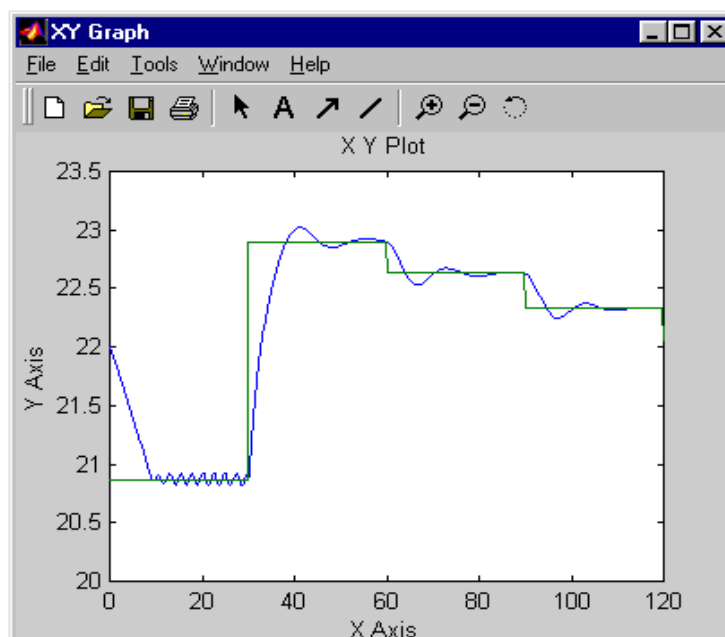


Рис.2.28. Графік регулювання концентрації продукту на виході реактора

2.7. Програмні пакети в області нейромережових технологій

Програму моделювання нейронної мережі називають програмою-імітатором або нейропакетом, розуміючи під цим програмну оболонку.

На ринку програмного забезпечення є безліч нейропакетів, що випускаються рядом фірм та окремими дослідниками, що дозволяють конструювати, навчати і використовувати нейронні мережі для вирішення практичних завдань.

Розглянемо деякі нейропакети, призначені для реалізації на персональних комп'ютерах.

1. *Пакет BrainMaker* виробництва американської компанії California Scientific Software з'явився на Заході в кінці 80-х рр. XX ст. і незабаром став одним з лідерів продажів. У 1990 році цей пакет отримав приз в номінації «Кращий програмний продукт року», а згодом став найбільш поширеним у США нейропакетом. Спочатку пакет був розроблений на замовлення NASA, а потім адаптований для комерційних додатків. Пакет набув широкого поширення в багатьох промислових і фінансових компаніях, в оборонних підприємствах США для вирішення задач прогнозування, оптимізації та моделювання ситуацій.

Пакет BrainMaker складається з двох компонентів: нейромережевого ядра, яке навчає і тестує нейронні мережі, і середовища для створення нейромереж, аналізу та підготовки вихідних даних. Пакет реалізує тільки одну парадигму: навчання з учителем методом зворотного поширення помилки, але цього було достатньо для більшості додатків в області класифікації і прогнозування.

2. *Пакет AITrilogy* американської фірми WardSystemsGroup, що представляє собою набір з трьох самостійних додатків: NeuroShell II, Neuro Windows, GeneHunter. NeuroShell II – це засіб створення, навчання і тестування нейромережових додатків, Neuro Windows – нейромережева бібліотека в початкових текстах, GeneHunter – система оптимізації мереж на основі генетич-

них алгоритмів. Разом ці компоненти утворюють потужний «конструктор», що дозволяє будувати аналітичні комплекси досить великої складності. Цей пакет також адаптований для бізнес-додатків і використовується у понад 150 найбільших банках США. У пакеті AI Trilogy є можливості обробки текстових даних, завдання правил у явному вигляді, роботи з фінансовими індикаторами й обробки циклічних подій.

3. *Пакет Statistica Neural Networks* відноситься до сучасних нейромережових продуктів і тому більш досконалий порівняно з раніше випущеними. Даний пакет – це універсальний пакет нейромережевого аналізу американської фірми Stat-soft. Він має потужні алгоритми навчання мережі (включаючи методи спряжених градієнтів і Левенберга-Маркара), можливість створення складних комбінацій з мереж різних архітектур. Для цього пакету характерні простота у використанні та аналітичні потужності, наприклад Automatic Network Designer (автоматичний конструктор мережі) визначить найкращу архітектуру для конкретного завдання, здійснить відбір змінних.

Для створення набору навчальних даних використовується вбудований редактор даних (File / New / DataSet), або завантажуються файл даних * .sta з диска. Після відкриття даних ST Neural Networks запропонує функцію автоматичної побудови (вікно Intelligent Problem Solver). На даному етапі необхідно відмовитися від даної функції (Cancel).

При відкритті нового файлу в ST Neural Networks всі змінні вважаються вхідними. Необхідно вказати вихідні змінні. Клацніть правою кнопкою миші в поле з назвою вихідних змінних у вікні Data Set Editor, у контекстному меню виберіть Output, колір заголовка стовпця зміниться на блакитний.

Всі спостереження необхідно поділити на дві множини: навчальну (служить для навчання НС) і контрольну (необхідна для оцінки ходу навчання).

У другому полі Cases вікна Data Set Editor вкажіть розмір контрольної множини (зазвичай на нього відводиться половина всього набору даних), натисніть Enter. Рядки, що містять контрольні спостереження, відмічені червоним кольором (знаходяться в кінці списку спостережень). Потім командою перемішати (Shuffle) (Edit → Cases → Shuffle → TrainandVerify), всі спостереження випадковим чином розподіляються за різними типами.

Для створення мережі використовується пункт меню File → New → Network. При цьому з'являється вікно редактора мережі (рис.2.29).

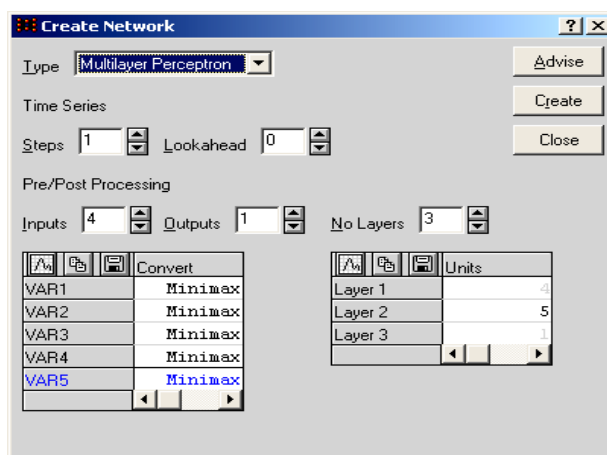


Рис. 2.29. Вікно редактора мережі

Тип мережі визначається назвою зі списку Type, Кількість шарів мережі задається в поле NoLayers. Кількість нейронів у шарі задається у вікні Units.

Мережа заданої структури і параметрами створюється після натискання кнопки Create.

Параметри Steps і Lookahead використовуються тільки у задачах часових рядів і в даній роботі не використовуються.

Навчання мережі.

Для мереж кожного типу використовуються спеціальні алгоритми навчання, які знаходяться в пункті меню Train. При виборі алгоритму пропонується визначити параметри навчання.

Наприклад, при навчанні багат шарового перцептрона методом зворотного поширення (Train → Multilayer Perceptrons → Back Propagation) задаються наступним (рис.2.30).

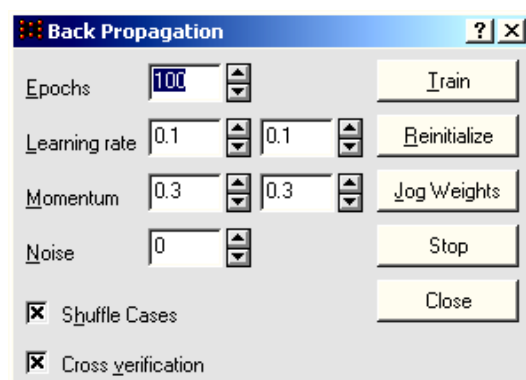


Рис.2.30. Вікно редактора параметрів навчання

Epochs – Епохи. Задає число циклів навчання, які проходяться при одному натисканні кнопки Train. Значення за замовчуванням 100.

Learningrate – Швидкість навчання, задає величину кроку при зміні ваг: при недостатній швидкості алгоритм повільно сходиться, при збільшенні швидкості навчання алгоритм працює швидше, але в деяких завданнях це може привести до нестійкості (особливо якщо дані зашумлені). Для швидкого і грубого навчання підійдуть значення від 0,1 до 0,6; для досягнення точної збіжності потрібні менші значення (наприклад, 0,01 або навіть 0,001, якщо епох багато тисяч). Іноді корисно зменшувати швидкість в процесі навчання.

Momentum – Інерція. Цей параметр покращує (прискорює) навчання в ситуаціях, коли помилка мало змінюється, а також надає алгоритму додаткову стійкість, допомагає алгоритму не застрягати в низинах і локальних мінімумах. Значення цього параметра завжди має лежати в інтервалі $[0; 1]$. Часто рекомендується використовувати високу швидкість навчання в поєднанні з невеликим коефіцієнтом інерції і навпаки.

Shuffle Cases – Перемішувати спостереження. При використанні цієї функції порядок, в якому спостереження подаються на вхід мережі, змінюється в кожній новій ітерації. Це додає в навчання деякий шум, так що помилка може відчувати невеликі коливання. Однак при цьому менше ймовірність того, що алгоритм «застрягне», і загальні показники його роботи зазвичай поліпшуються.

Cross-verification (крос-перевірка) – стандартний спосіб навчання нейронних мереж полягає в тому, що мережа навчається на одному з множин, а на іншому перевіряється результат; таким чином, контрольна множина для навчання не використовується. Це дає незалежний спосіб перевірки того, чи навчилася мережа чомусь корисному.

Reinitialize – скидання параметрів мережі, що настраюються, отриманих на попередньому циклі навчання, або для попередньої мережі.

Запуск алгоритму навчання проводиться натисненням кнопки Train. Кожен раз при натисканні кнопки Train алгоритм робить N циклів навчання, де N – кількість циклів навчання, яка визначається в Epochs.

На графіку (Statistics → Training Error Graph) можна спостерігати зміну помилки мережі в ході навчання. Training Error Graph – це відображення середньоквадратичної помилки виходу на всьому навчальній множині.

У пакеті STATISTICA Neural Networks передбачено автоматичне запам'ятовування кращої мережі під час експерименту. Для відновлення значень найкращою мережі викличте меню Train → Auxiliary → Best Network.

Для перегляду результатів навчання мережі використовується функція Options у вікні Network Set Editor, у вікні потрібно натиснути Add (додати). При цьому у вікні Network Set Editor додається інформаційна рядок для мережі: тип мережі, середньоквадратична помилка мережі (RMSError),

кількість входів мережі, число прихованих елементів в мережі, коефіцієнт регресії, використані методи навчання (якщо в списку Detailshown вибрано Verbose – детальний).

Для розрахунку даних за допомогою мережі використовується пункт меню Run:

DataSet – розрахунок для всіх даних вихідного файлу;

SingleCase – розрахунок одного спостереження;

One-off – розрахунок для довільного вхідного вектора.

Розрахунок проводиться при натисканні кнопки Run відповідного вікна.

Запускається як і в попередньому випадку, але в таблицях виводиться тільки фактичне вихідне значення.

Відкрийте меню *Run* → *One-off*, введіть вхідні значення, для яких необхідно спрогнозувати вихідний, натисніть Run.

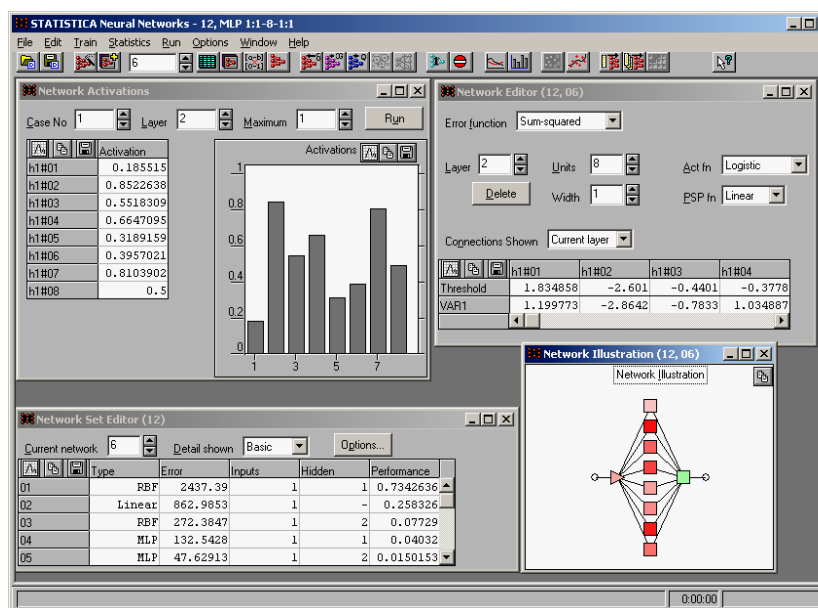


Рис.2.31. Вікна перегляду параметрів навченої мережі

Перегляд вагових коефіцієнтів синаптичних зв'язків нейронів можливий у вікні Network Editor (рис.2.31) (Edit → Network ...). Значення Theshold – порогове значення, яке віднімається від вхідного значення нейрона. Значення VAR1 (на рис. 2.31) – ваго-

вий коефіцієнт зв'язку. Номер шару задається в поле Layer. У даному вікні можна переглянути (задати) функцію активації нейрона кожного шару (поле Actfn).

Перегляд вихідних значень нейронів у шарі у вікні Network Activations (Run → Activations ...). Розрахунок для рядка даних (вказана в полі CaseNo) після натискання кнопки Run.

У пакеті STATISTICA Neural Networks передбачена можливість автоматичного визначення структури кращої мережі для набору навчальних даних (функція доступна через File → New → Intelligent Problem Solver).

4. Програмний пакет Neural Network Wizard.

Шлях до програми: NeuralNetworkWizard \ Bin \ Wizard.exe.

Перш, ніж запускати програму, необхідно в будь-якому текстовому редакторі підготувати текстовий файл з навчальною вибіркою. Приклад такого файлу для функції $res=s_1+s_2$ наведено нижче.

s1	s2	res
0	0	0
1	1	2
2	2	4
3	3	6
4	4	8
5	5	10
6	6	12

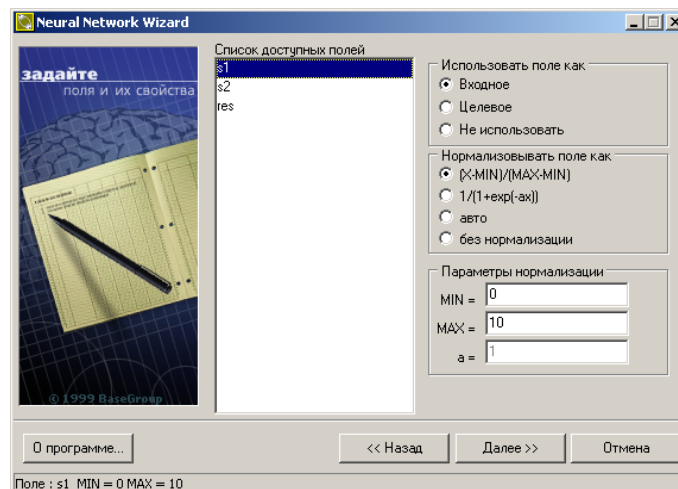


Рис.2.32. Список доступних полів

У першому рядку файлу вказуються імена вхідних/вихідних змінних: s_1 і s_2 – імена вхідних змінних, res – ім'я вихідної змінної. Далі йдуть їх значення в колонках. Файл зберігається як текстовий з розширенням ТХТ. Після запуску програми в першому вікні задається ім'я файлу. Кнопка «Далі» дозволяє перейти до наступного вікна (рис.2.32).

У новому вікні перерахований список доступних полів, взятий з першого рядка зазначеного файлу. Для кожного з полів необхідно вказати, чим є дана змінна. Якщо дана змінна вхідна, то в групі «Використовувати поле як» вибирається варіант «Вхідний», якщо вихідна – «Цільове». Крім того, для кожної змінної можна вказати вид нормалізації (приведення до діапазону $[0, 1]$) з відповідними параметрами. Кнопка «Далі» переводить до наступного вікна.

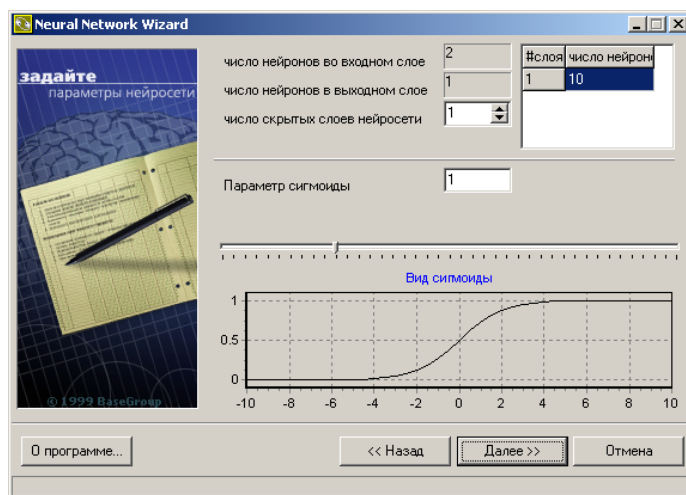


Рис.2.33. Структура і параметри НМ

У даному вікні (рис.2.33) визначаються структура і параметри НС: кількість прихованих шарів, кількість нейронів у кожному шарі, а також вид сигмоїдної функції.

У наступному вікні (рис.2.34) задаються параметри навчання і критерії зупинки навчання, якщо вона потрібна. Кнопка «Далі» показує короткий попередній звіт.

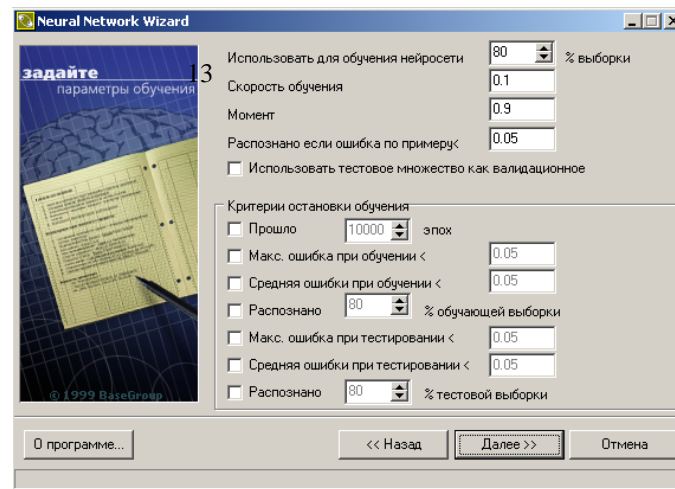


Рис.2.34. Параметри навчання і критерії зупинки навчання

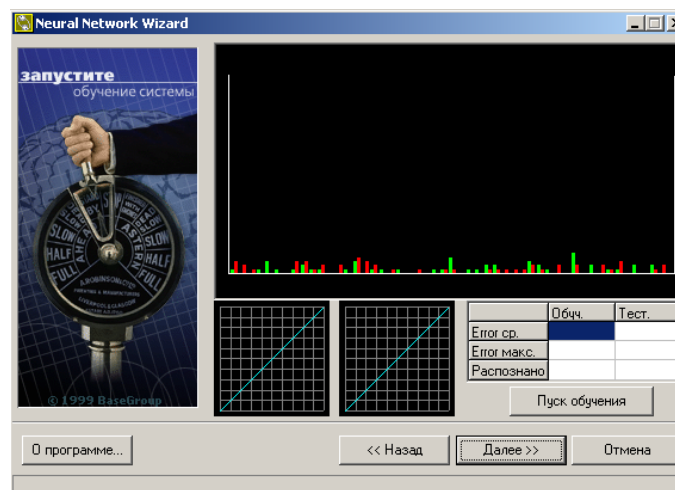


Рис.2.35. Візуалізація процесу навчання

У наступному вікні візуалізований процес навчання (рис. 2.35). Щоб запустити навчання, натискається кнопка «Пуск навчання». На верхній діаграмі показано розподіл помилки навчання: по горизонталі значення помилки (чим правіше стовпець, тим більше помилка), по вертикалі кількість прикладів з вибірки з даною помилкою. Зелені стовпці – помилка на робочій навчальній вибірці, червоні – на тестовій. У процесі навчання стовпці повинні прагнути в ліву частину діаграми.

Нижче діаграми відображається розподіл прикладів на робочій і тестовій вибірках. Кожен приклад зображений тут точкою. Чим ближче точка до діагоналі, тим точніше НС передбачила її значення.

Зупинка навчання відбувається або за раніше вказаним критерієм, або за допомогою тієї ж кнопки «Пуск навчання».

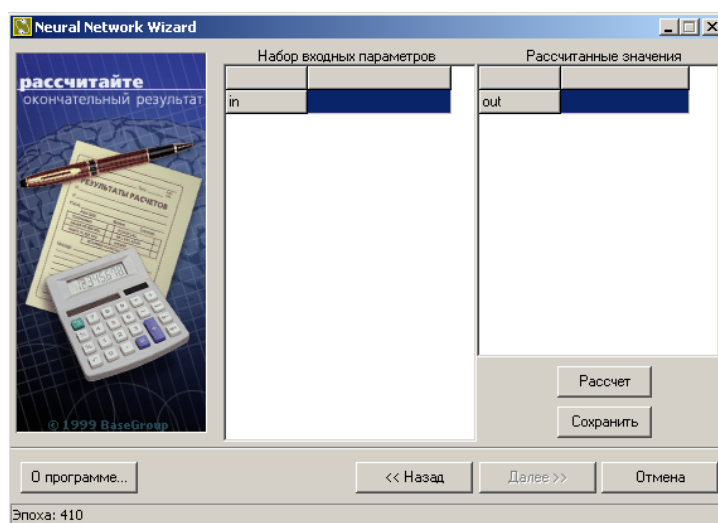


Рис.2.36. Точність роботи НМ

У наступному вікні (рис.2.36) надають можливість оцінити точність роботи НС в експлуатаційному режимі. Для цього в лівому полі вказується довільне значення вхідного сигналу. Після натискання на кнопку «Розрахунок» у правому полі з'являється розраховане мережею значення вихідного.

Кнопка «Зберегти» дозволяє записати параметри навченої мережі у вигляді файлу (за замовчуванням розширення файлу NNW).

У даному файлі крім інших параметрів указані: Epoch – кількість епох (циклів) навчання; Layer * – кількість нейронів у відповідному шарі (нейрони нумеруються від 0 до N-1), ваги синапсів (i = номер шару – 2, j – номер нейрона, k – номер синапсу даного нейрона).

5. *NeuroPro*. Програма NeuroPro створена у 98-му році, але не втрачає своєї актуальності і до цього дня. Вона успішно працює і на Windows XP, і на Vista, і на 7-ці. Версія 0.25 поширюється безкоштовно, її можна знайти на багатьох сайтах в Інтернеті. NeuroPro вміє будувати багат шарові нейронні мережі з сігмоїдною функцією активації. Нейромережу можна навчити на одному масиві даних, а потім протестувати на іншому. Вбудовані можливості нейроіміатора для редагування файлів даних досить обмежені. Для внесення змін до файлу даних доцільно скористатися табличним процесором MS Excel з подальшим збереженням файлу в форматі DBF 4, який читається програмою Neuropro 0.25. Додавання в нейропроект нейронної мережі шаруватої архітектури з числом шарів нейронів від 1 до 10, числом нейронів у шарі – до 100 (число нейронів для кожного шару мережі може задаватися окремо). Для створення нової мережі необхідно клацнути по кнопці «Нова мережа», яка знаходиться за таблицею даних і стає активною після відкриття файлу даних.

Навчання нейронної мережі на певному задачнику проводиться градієнтними методами оптимізації, градієнт обчислюється за принципом подвійності. У програмі реалізовані чотири алгоритми оптимізації:

- метод найшвидшого спуску;
- модифікований ParTan;
- метод сполучених градієнтів;
- квазіньютонівський BFGS-метод.

При створенні нейропроекту як алгоритму, за замовчуванням приймається ParTan. Зміна алгоритму здійснюється через пункти меню «Налаштування» – «Метод оптимізації».

Прикладу задачника відповідає запис (рядок) файлу даних. Для включення запису файлу даних до задачника, у записі повинні міститися дані для всіх полів, використовуваних нейронною мережею в якості вхідних і вихідних. Навчання припиняється

ся при досягненні заданої точності рішення задачі або при неможливості подальшої оптимізації.

Нейромережа може вирішувати одночасно кілька завдань – як прогнозування (передбачення значень декількох кількісних ознак), так і завдання класифікації (передбачення станів кількох якісних ознак), так і завдання прогнозування і класифікації одночасно для кожної із задач можуть бути встановлені свої вимоги до точності.

На даний момент є можливість роботи тільки з перехідним і шаруватими монотонними нейронними мережами з числом шарів нейронів від 1 до 10, числом нейронів в шарі – до 100. Число нейронів у шарі не залежить від числа вхідних сигналів і числа вихідних сигналів. Після останнього шару нейронів мережа має шар адаптивних суматорів з числом суматорів, що дорівнює числу вихідних сигналів, з яких і знімаються вихідні сигнали мережі. Ваги синапсів при навчанні можуть змінюватися у діапазоні $[-1, 1]$, при створенні мережі ініціюються випадковими числами.

Сьогодні нейрокомп'ютерний сегмент ринку програмного забезпечення бурхливо розвивається, і поява нових пакетів є природним процесом. Більшість фірм, що працюють на ринку аналітичних програм, вже заявили про підготовку або випуск систем на основі нейромережевої технології. У сучасних нейропродуктів поліпшений інтерфейс користувача, включені додаткові алгоритми навчання мереж, реалізовані можливості взаємодії з іншими додатками. Однак плата за ці можливості – необхідність застосування більш потужного комп'ютера, зростання вимог до обсягу оперативної і дискової пам'яті. Необхідно також врахувати, що новий пакет зовсім не гарантує більш якісного вирішення завдання користувача.

РОЗДІЛ 3

НЕЧІТКЕ МОДЕЛЮВАННЯ

Останнім часом нечітке моделювання є одним з найбільш активних і перспективних напрямків прикладних досліджень у сфері управління й прийняття рішень [20]. Нечітке моделювання стає особливо корисним, коли в описі технічних систем і бізнес-процесів є невизначеність, яка ускладнює або навіть унеможливорює застосування точних кількісних методів і підходів.

Аналіз конкретної системи фактично полягає у виділенні певної підсистеми з більш великої системи. Виділяючи підсистему, ми фактично вводимо границі, яких насправді не існує. При побудові моделей реальних явищ дослідник, бажаючи користуватися математичними методами, часто вносить визначеність там, де її немає за існуючими умовами. Особливо небезпечним є використання штучної визначеності при управлінні.

Не маючи можливості точно описати всі підсистеми зв'язків з системою, ми використовуємо або власні уявлення про ці зв'язки, або звертаємося за допомогою до експертів, які цими уявленнями володіють. Ці уявлення або, інакше, інформація про межі аналізованої підсистеми, найчастіше бувають виражені в поняттях, які мають нечіткий сенс з точки зору класичної математики.

Мова традиційної математики, що спирається на теорію множин і двозначну логіку, виявляється недостатньо гнучкою для моделювання реальних складних систем, оскільки в ній немає засобів описання понять, якими користується людина і які мають невизначений сенс. Чим складніше система, тим глибше ми аналізуємо завдання, тим більш невизначеним стає її рішення. Простий приклад – класифікація об'єктів за кольором. У традиційній математиці класифікація – це розбивання деякої

сукупності об'єктів на непересічні підмножини. Проте ми не в змозі провести межу між окремими кольорами.

Одна з основних цілей побудови математичних моделей систем – обробка наявної інформації для вибору раціонального (оптимального) варіанта управління системою. Дуже часто, особливо при дослідженні економічних і соціальних систем, інформація про систему носить суб'єктивний характер і, як правило, містить велику кількість невизначеностей типу "багато", "мало", "сильно", "слабко", "високий", "низький" тощо, які не мають аналогів у мові традиційної математики. За глибоко вкоріненою традицією наукове мислення розуміння явища, на жаль, ототожнюється з можливістю його кількісного аналізу. Наявність математичних засобів відображення нечіткості вихідної інформації дозволило б побудувати модель, адекватну реальності.

Необхідність успішного використання математичних методів для аналізу складних систем призвела до створення засобів більш точного і повного урахування нечітких уявлень і суджень людей про реальний світ у математичних моделях. Було створено новий напрямок у галузі прикладної математики, пов'язаний з ім'ям видатного американського математика Лотфі Заде, який отримав назву теорії нечітких множин. Л. Заде сформулював принцип несумісності: чим складніша система, тим менше ми здатні дати точні і в той же час практичні судження про її поведінку.

Для систем, складність яких перевищує деякий пороговий рівень, точність і практичний сенс стають практично виключними один для одного характеристиками.

В області управління технічними системами нечітке моделювання дозволяє отримувати адекватні результати порівняно з результатами, які ґрунтуються на використанні традиційних аналітичних моделей та алгоритмів управління. Діапазон застосування нечітких методів з кожним роком

розширюється, охоплюючи такі галузі, як проектування промислових роботів і побутових електроприладів, управління доменними печами і рухом поїздів метро, управління промисловою безпекою підприємств, автоматичне розпізнавання мови і зображень.

Нечітка логіка, яка служить основою для реалізації методів нечіткого управління, більш природно описує характер людського мислення і хід його міркувань, ніж традиційні формально-логічні системи. Саме тому вивчення та використання математичних засобів для представлення нечіткої вихідної інформації дозволяє будувати моделі, які найбільш адекватно відображають різні аспекти невизначеності, постійно присутні в реальності.

Вкажемо основні причини, на основі яких віддається перевага застосуванню системи з нечіткою логікою:

- 1) концептуально легша для розуміння;
- 2) гнучка система стійка до неточних вхідних даних;
- 3) може моделювати нелінійні функції довільної складності;
- 4) у ній ураховується досвід фахівців-експертів;
- 5) заснована на природній мові людського спілкування.

Успіх нового підходу визначається його прикладною спрямованістю. Перш за все, це не суха теорія, а апарат вирішення великого класу практичних завдань. Це і теорія алгоритмів і штучний інтелект, теорія прийняття рішень, розпізнавання образів, "препарування" природної мови (визначення нечіткого сенсу виразів) і т.ін. Теорія нечітких множин – єдина теорія, яка математично оперує зі смисловим змістом слів людини (математична лінгвістика).

Поняття нечіткої множини, що лежить в основі цієї теорії, є способом математичного моделювання невизначених понять, якими оперує людина при описі своїх уявлень про реальні системи, своїх бажань, цілей і т.ін. **Нечіткі множини – це математична модель класу з нечіткими (розмитими)**

межами. У цьому понятті враховується можливість поступового переходу від приналежності до неналежності) елемента множині, тобто елемент може мати ступінь приналежності множини, проміжну між повною приналежністю і повної неналежністю.

Теорія нечітких множин являє собою узагальнення і переосмислення найважливіших напрямів класичної математики. Біля її витоків лежать, зокрема, ідеї та досягнення багатозначної логіки, яка вказала на можливість переходу від двох до довільного числа значень істинності і поставила проблему оперування поняттями з мінливим змістом.

Заде показав, яким чином нечітку, якісного характеру інформацію можна використовувати у формалізованих процедурах аналізу. Було запропоновано таке розширення мови математики, яке дозволяє враховувати нечіткість вихідної інформації в математичних моделях.

В даний час у сфері бізнесу та економіки широко використовуються людино-машинні системи, призначені для обробки нечітких даних, що реалізують і надають допомогу у прийнятті економічних рішень, маркетингу, поради по вкладенню капіталу, різного роду управління і планування, аналіз ризику, допомогу у підготовці контрактів і т.ін.

Для кращого розуміння багатьох реальних управлінських ситуацій у даний час необхідне володіння засобами конструктивного аналізу ситуацій, використання засобів їх моделювання. Основа моделювання – математика. Але часто управлінські ситуації, що вимагають прийняття рішень, настільки складні й невизначені (особливо на початкових етапах аналізу), що їх не вдається описати, використовуючи апарат класичної математики. У багатьох управлінських ситуаціях класична математика просто "не працює".

3.1. Методологія нечіткого моделювання

Насамперед слід зауважити, що методологія нечіткого моделювання не замінює і не виключає розглянуту вище методологію системного моделювання (рис.3.1, [21]), а конкретизує останню стосовно до процесу побудови і використання нечітких моделей складних систем.

Процес нечіткого моделювання являє аналогічну послідовність взаємопов'язаних етапів, як і процес системного моделювання. При цьому кожен з етапів виконується з метою побудови і використання нечіткої моделі системи вирішення вихідної проблеми.

У загальному випадку під нечіткою моделлю розуміється інформаційно-логічна модель системи, побудована на основі теорії нечітких множин і нечіткої логіки.

У рамках сучасної методології системного моделювання невизначеність може характеризувати такі аспекти модельних уявлень:

1. Неясність або нечіткість межі системи. Так, наприклад, використання дихотомічних ознак "великий–малий", "високий–низький", "дорогий–дешевий", "молодий–старий", "досвідчений–недосвідчений", "швидкий–повільний" і подібних їм для визначення складу елементів системи стикається з принциповою перешкодою подання структури моделі системи. Характерний приклад цього аспекту невизначеності – власне клас складних систем в контексті відповіді на питання: "Які системи слід вважати складними?"

2. Неоднозначність семантики окремих термінів, які використовуються при побудові концептуальних моделей систем (модель зачіски та математична модель, гральний автомат та автомат як стрілецька зброя, графічна карта місцевості, карта тощо).

3. Неповнота модельних уявлень про деяку складну систему, особливо у зв'язку з рішенням слабо формалізованих проблем. У цьому випадку сама спроба побудувати адекватну модель складної системи або предметної області стикається з принциповою неможливістю врахувати всі особливості розв'язуваної проблеми.

4. Суперечливість окремих компонентів модельних уявлень або вимог, яким повинна задовольняти модель складної системи. Так, наприклад, вимога вирішити проблему за мінімальний час і з мінімальними фінансовими витратами містить в собі елемент протиріччя.

5. Невизначеність настання тих або інших подій, що відносяться до можливості знаходження системи-оригіналу в тому чи іншому стані в майбутньому. Мова йде про те, що аналіз процесу поведінки системи не дає підстави для однозначної відповіді на питання: "чи буде знаходитися система-оригінал в деякому стані в момент часу, який відноситься до її майбутнього?" Цей аспект невизначеності часто називають стохастичним, оскільки він традиційно досліджувався засобами теорії ймовірностей і математичної статистики.

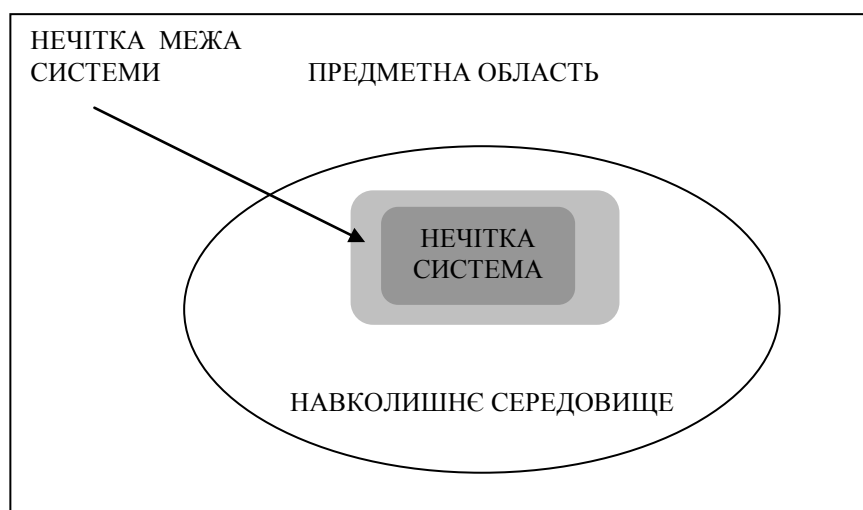


Рис. 3.1. Графічна ілюстрація нечіткої системи як системи з нечіткою межею

Таким чином, нечітка модель системи-оригіналу, або нечітка система в першу чергу характеризується невизначеністю типу неясності (нечіткості) межі системи, а також, можливо, окремих її станів, вхідних і вихідних впливів. У цьому випадку вихідна структуризація нечіткої системи може бути зображена графічно у вигляді фігури з розпливчастими межами (рис. 3.1).

3.2. Математичні основи нечітких систем

Поняття нечітких множин (fuzzysets) як узагальнення звичайних (чітких) множин було введено Лотфітом Заде у 1965 р. [7]. Про множини відомо, як мінімум, що вона складається з елементів, які мають деяку загальну властивість. Надалі множини будемо позначати великими літерами латинського алфавіту, а елементи множини – малими літерами. Належність елемента a до множини M позначається як: $a \in M$ неналежність $a \notin M$ або $a \bar{\in} M$. Множина A є підмножиною множини B (позначається $A \subseteq B$), якщо будь-який елемент з A є елементом B . Якщо при цьому $A \neq B$, то A називається суворою (власною) підмножиною множини B (позначається $A \subset B$).

Множина, для якої всі розглядувані множини є підмножинами, називається універсальною множиною (універсумом). Універсальна множина, таким чином, представляє повний простір, область міркувань (X, Y, Z, U, V – універсальні множини; A, B, C, \dots – підмножини універсальних множин; $a, b, c, \dots, x, y, z, u, v$ – елементи відповідних множин).

Способи завдання множин:

1. Перерахуванням (списком) елементів – тільки для кінцевих множин. Наприклад: $A = \{a, b, c, d\}$.

2. Описом властивостей, якими повинні володіти її елементи. Це позначається як $M = \{x | P(x)\}$ або $M = \{x : P(x)\}$ і читається: множина M складається з таких елементів x , що x має властивість P .

3. Породжується процедурою, що описує спосіб отримання елементів. Наприклад, нових елементів b множини M з вже існуючих елементів a : $M = \{b: b = (a+1)^2, a \in N\}$ звідки при $a = 1, 2, 3$ отримуємо набір $M = \{4, 9, 16 \dots\}$.

4. За допомогою характеристичної функції. Цей спосіб особливо важливий з точки зору поняття нечіткої множини.

Традиційний спосіб подання елемента множини A полягає в застосуванні характеристичної функції. *Характеристична функція* χ_A , визначає множину B повному просторі X , являє собою відображення, для якого $X \in$ областю визначення, а $\{0, 1\}$ (двозначна множина з 0 і 1) область значень:

$$\chi_A(x) = \begin{cases} 0, & x \notin A \\ 1, & x \in A \end{cases}$$

тобто $\chi_A(x) = 1$, якщо елемент x задовольняє властивостям A , і $\chi_A = 0$, якщо не задовольняє. Якщо відкласти значення X на горизонтальній осі, а $\{0, 1\}$ на вертикальній, то отримаємо графічне представлення, показане нарис. 3.2.

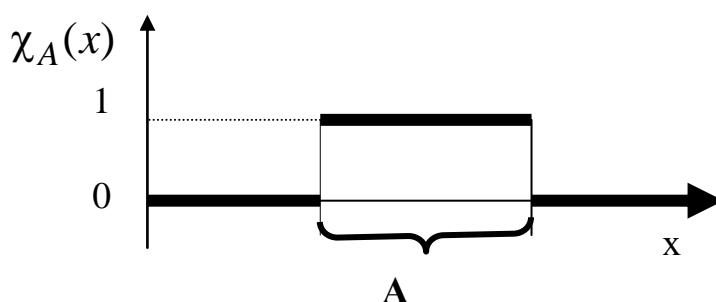


Рис. 3.2. Графічне подання характеристичної функції

В нечітких системах елемент може частково належати до будь-якої множини. Ступінь приналежності до множини A , що представляє собою узагальнення характеристичної функції, називається функцією належності $\mu_A(x)$, причому $\mu_A(x) \in [0, 1]$.

У традиційній (класичній) математиці множина розуміється як сукупність елементів (об'єктів), що володіють деякою

загальною властивістю. Для будь-якого елемента при цьому розглядаються лише дві можливості: або цей елемент належить даній множині (тобто володіє даною властивістю), або не належить даній множині (тобто не володіє даною властивістю).

Однак, при спробах математичного опису ряду систем мова звичайних множин може виявитися недостатньо гнучкою. Часто наявна інформація про систему формулюється мовою нечітких понять, які неможливо формалізувати за допомогою звичайних множин.

Припустимо, що деяке явище, властивість, подія, наприклад якість функціонування деякої системи оцінюється параметром (показником), причому A може приймати ряд нечітких (розпливчастих) значень:

$$\begin{aligned}x_1 &= \{\text{працює відмінно}\}, \\x_2 &= \{\text{працює дуже добре}\}, \\x_3 &= \{\text{працює добре}\}, \\x_4 &= \{\text{працює погано}\}, \\x_5 &= \{\text{не працює}\},\end{aligned}$$

Поняття нечіткої множини – спроба математичної формалізації нечіткої інформації з метою її використання при побудові математичних моделей систем. В основі цього поняття лежить уявлення про те, що елементи цієї множини, які володіють загальною властивістю, можуть володіти цією властивістю в різній мірі і, відповідно, належати даній множині з різним ступенем. При такому підході висловлювання типу: "елемент x належить до даної множини A ($x \in A$)" втрачає сенс, оскільки необхідно вказати наскільки "сильно" або з яким ступенем даний елемент x належить даній множині A .

Один з найпростіших способів математичного опису нечіткої множини – характеристика ступеня приналежності елемента до множини A числом, наприклад, з інтервалу $[0,1]$.

Побудова заходів приналежності може бути пов'язана з експертним підходом.

Повернемося до прикладу оцінки якості функціонування деякої системи. При формалізації розпливчастості кожному значенню приписується певне число p_i , яке кількісно оцінює міру того, що параметр A приймає значення $x_i (i = 1, \dots, n)$. Нехай для цього залучається група з N рівноправних і незалежних експертів, кожен з яких має рівно n голосів. Кожен експерт, оцінюючи той факт, що параметр A приймає значення x_i може віддати один голос значенням x_i , а може і не віддати, причому один експерт може віддати по одному голосу одразу декільком. У цьому проявляється *нечіткість знання експертів*. Максимальна кількість голосів, яку може зібрати значення x_i дорівнює N , і це в тому випадку, коли всі експерти віддали по одному голосу значенням x_i .

Нехай значення x_i набрало N_i голосів. Тоді в якості кількісної оцінки того факту, що параметр A набуває значення x_i , приймемо

$$p_i = \frac{N_i}{N}, \quad i = 1, \dots, n, \quad 0 \leq p_i \leq 1.$$

Формалізуємо дані розглянутого прикладу. Нехай X – деяка множина (звичайна): $X = \{x_1, x_2, \dots, x_n\}$. Зв'яжемо з параметром A *нечітка підмножина* A (частіше використовується термін "нечітка множина"), під яким розуміють сукупність пар виду (x_i, p_i) , де число p_i визначає ступінь приналежності елемента x_i нечіткій множині A . Функція $\mu_A: X \rightarrow [0,1]$ визначається за правилом $\mu_A(x_i) = p_i$, називається *функцією належності* нечіткої множини A . Значення p_i цієї функції для конкретного елемента x_i називається *ступенем належності цього елемента* нечіткій множині A . Універсальна множина X нечіткої множини A розуміється як область визначення функції приналежності μ_A . Таким чином, область визначення нечіткої множини - звичайна

(чітка) множина. Як видно з цього визначення, нечітка множина повністю визначається своєю функцією приналежності, тому функцію приналежності часто використовують як позначення нечіткої множини. Значення функції є раціональними числами з інтервалу $[0,1]$, де 0 означає відсутність належності до множини, а 1 – повну приналежність. Конкретне значення функції приналежності називається ступенем або *коефіцієнтом приналежності*. Цей ступінь може бути визначеним явним чином у вигляді функціональної залежності, наприклад,

$$\mu_A(x) = \exp \left[- \left(\frac{x-3}{0,2} \right)^2 \right]$$

або дискретно – шляхом завдання кінцевої послідовності значень x . Існують різні форми запису нечітких множин. Наприклад, кінцеву нечітку множину можна записати у вигляді:

$$A = \{x_1 | p_1, x_2 | p_2, \dots, x_n | p_n\}$$

або

$$A = \{x_1 | \mu_A(x_1), x_2 | \mu_A(x_2), \dots, x_n | \mu_A(x_n)\}$$

або

$$A(x) = \left\{ \frac{x_1}{\mu(x_1)}, \frac{x_2}{\mu(x_2)}, \dots, \frac{x_n}{\mu(x_n)} \right\}, A(x) = \left\{ \frac{\mu(x_1)}{x_1}, \frac{\mu(x_2)}{x_2}, \dots, \frac{\mu(x_n)}{x_n} \right\}.$$

Наприклад, для послідовності дискретних значень змінної x , рівних $x_1 = 7, x_2 = 8, x_3 = 9, x_4 = 10, x_5 = 11, x_6 = 12, x_7 = 13$, їх коефіцієнт приналежності до чисел, близьким 10, може бути визначений у вигляді

$$A(x) = \left\{ \frac{0,1}{7}, \frac{0,3}{8}, \frac{0,8}{9}, \frac{1,0}{10}, \frac{0,8}{11}, \frac{0,3}{12}, \frac{0,1}{13} \right\}.$$

Одноточкова нечітка множина містить один елемент:

$$A = \mu_A(x) | x.$$

Будь нечітку множину можна розглядати як об'єднання складових її одноточкових множин.

Визначення. Нечітка підмножина A універсальної множини X характеризується функцією приналежності $\mu_A : X \rightarrow [0,1]$, яка ставить у відповідність кожному елементу числа з інтервалу $[0,1]$, що характеризує ступінь приналежності елемента x підмножини A . Форма запису нечіткої підмножини A універсальної множини X має вигляд

$$A = \mu_1 / x_1 + \mu_2 / x_2 + \dots + \mu_n / x_n \text{ або } A = \sum_{i=1}^n \mu_i / x_i,$$

де μ_i – ступінь приналежності елемента $x_i \in X$ нечіткій підмножині A універсальної множини X ; знак підсумовування позначає об'єднання одноточкових множин μ_i / x_i . У разі, якщо задано безперервну функцію μ_i , нечітка підмножина A універсальної множини X представляється у вигляді

$$A = \int_X \mu_A(x) / x,$$

де символ інтеграла позначає операцію об'єднання одноточкових множин μ_A / x , а не підсумовування.

Наведене визначення є основою для формалізації різного виду невизначеностей.

Приклад 3.1. Нехай діапазон зміни параметра визначається універсальною множиною $U = 0 + 1 + 2 + 3 + 4 + 5$, а значення величини параметра при спостереженні характеризують нечітким терміном "високий". На етапі формалізації якісної інформації термін "високий" зіставляється з нечіткою підмножиною A універсальної множини U . Припустимо, що це зіставлення дало результат:

$$A = \text{високий} = 0/0 + 0,1/1 + 0,2/2 + 0,5/3 + 0,8/4 + 1/5.$$

Даний вираз показує, що значення величини параметра $u_1 = 0$ не може бути віднесено до поняття "високий", тому

$\mu_1(u_1) = 0$. Величина параметра $u_4 = 3$ зі ступенем $\mu(u_4) = 0,5$ віднесена до поняття "високий", а для ступінь приналежності до даного терміну прийнята рівною 1. Останнє показує, що значення величини параметра $u_6 = 5$ при заданій універсальній множині U строго відповідає терміну "високий".

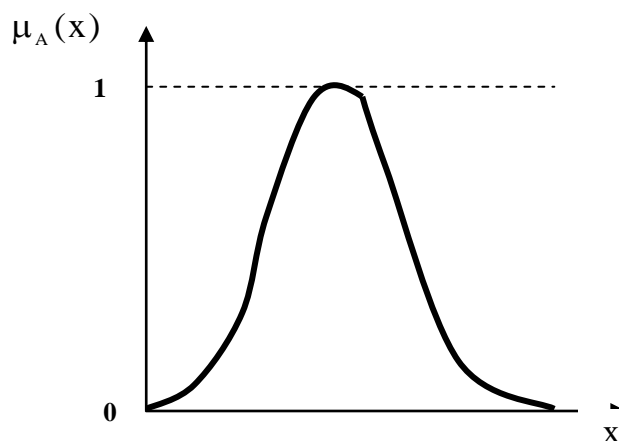


Рис.3.3. Графік функції приналежності

Нечітка множина, таким чином, формується шляхом введення узагальненого поняття приналежності, тобто розширення двохелементної множини характеристичної функції $\{0,1\}$ до континууму $[0,1]$. Це означає, що перехід від повної приналежності об'єкта класу до повної його приналежності відбувається не стрибком (рис. 3.2), а плавно, поступово (рис. 3.3).

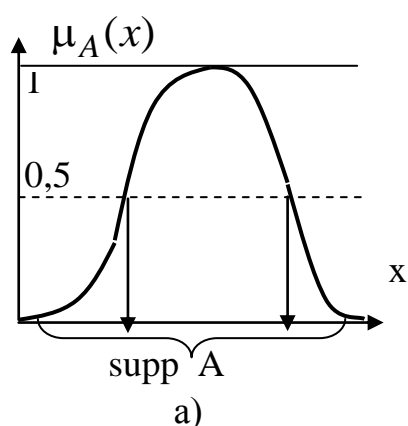
Звичайні множини при цьому складають підклас класу нечітких множин. Функцією приналежності звичайної множини $B \subset X$ є його характеристична функція

$$\chi_B(x) = \begin{cases} 0, & \text{якщо } x \notin B \\ 1, & \text{якщо } x \in B, \end{cases}$$

тобто звичайну (чітку) множину можна визначити як сукупність пар виду $(x, \mu_B(x))$, де $\mu_B(x)$ приймає тільки два значення: 0 і 1. Таким чином, нечітка множина являє собою більш широке поняття, ніж звичайне множина, в тому сенсі, що функція

належності нечіткої множини може бути довільною функцією та приймати значення в інтервалі $[0,1]$.

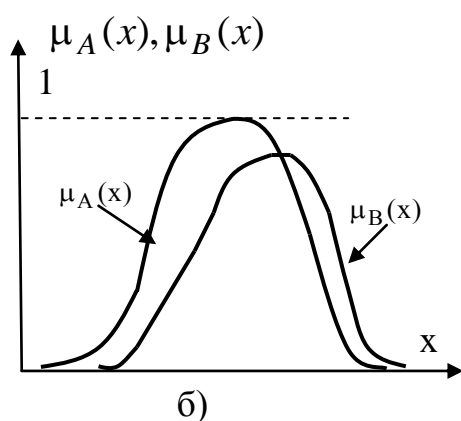
Якщо в експертній оцінці для кожного $x_i \in X$ виконано рівність $p_i = 0$, то кожному x_i жоден з експертів не віддав голоси, тобто з точки зору експертів явище, властивість або подія A не має місця. У цьому випадку нечітка множина A буде порожньою \emptyset : $\mu_{\emptyset}(x) = 0, \quad \forall x \in X$. (\forall – квантор спільності, для всіх x).



Точкою переходу нечіткої множини A називається такий елемент $x \in X$, для якого $\mu_A(x) = 0,5$ (рис. 3.4,а).

Нечітка множина називається нормальною, якщо верхня межа його функції приналежності досягає значення, рівного 1 (множина A на рис. 3.4,б):

$$\sup_{x \in X} \mu_A(x) = 1.$$



В іншому випадку множину називають субнормальною (множина на рис. 3.4, в). Нехай A і B – нечіткі множини X , а $\mu_A(x)$ і $\mu_B(x)$ – їх функції приналежності. Кажуть, що множина A включає в себе множину B (рис. 3.4,в) ($B \subseteq A$), якщо для будь-якого $x \in X$ виконується нерівність $\mu_B(x) \leq \mu_A(x)$. Таким чином, універсальну множину X можна описати функцією приналежності.

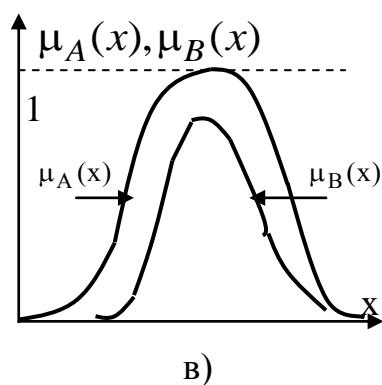


Рис. 3.4. Точка переходу і нормальна нечітка множина

При цьому $\sup p \quad B \subseteq \sup p \quad A$ (рис. 3.4, в). Множини A і B збігаються (еквівалентні), якщо $\mu_A(x) = \mu_B(x)$ при будь-якому $x \in X$.

Носієм нечіткої множини A (позначається $\text{supp } A$) з функцією приналежності $\mu_A(x)$ називається множина в звичайному сенсі для (рис. 3.4, а): $\sup p \quad A = \{x \mid x \in X, \mu_A > 0\}$.

Іншими словами, носієм нечіткої підмножини A називається множина елементів $x \in X$, для яких функція ступенів належності $\mu_A(x) > 0$.

Приклад 3.2. Нехай, наприклад, універсальна множина розглянутих контрактів складається з 4 елементів:

$$X = \{x_1, x_2, x_3, x_4\}.$$

Нечітка множина:

$$A = (\text{вигідний контракт}) = \{x_1 \mid \text{мало}, x_2 \mid \text{середньо}, x_3 \mid \text{сильно}, x_4 \mid \text{середньо}\}.$$

Тут нечіткі ступені приналежності: мало, середньо, сильно є нечіткими підмножинами допоміжної універсальної множини V , визначеної, наприклад, так чином:

$$V = \{0; 0,1; 0,2; 0,3; \dots, 0,9; 1\}.$$

При цьому нечіткі ступені приналежності визначаються, наприклад, так:

$$\text{мало} = \{0,2 \mid 0,5; 0,3 \mid 0,7; 0,4 \mid 1; 0,5 \mid 0,5\};$$

$$\text{середньо} = \{0,5 \mid 0,7; 0,6 \mid 1; 0,7 \mid 0,7; 0,8 \mid 0,5\};$$

$$\text{сильно} = \{0,7 \mid 0,5; 0,8 \mid 0,8; 1 \mid 1\}.$$

Вибір виду функцій належності та їх параметрів визначаються більшою мірою досвідом, інтуїцією та іншими суб'єктивними факторами особи, що приймає рішення.

В теорії нечітких множин, крім змінних цифрового типу, існують лінгвістичні змінні зі значеннями, що їм приписують.

Нехай змінна x позначає температуру ($x = \text{"температура"}$). Можна визначити нечіткі множини "негативна", "близька до нуля", "позитивна", характеризуються функціями належності:

$$\mu_{\text{негат.}}(x), \mu_{\text{бнул}}(x), \mu_{\text{позит.}}(x).$$

Так само як звичайна змінна може приймати різні значення, лінгвістична змінна "температура" може приймати різні лінгвістичні значення. У даному прикладі це: "негативна", "близька до нуля" і "позитивна". Отже, лінгвістичний вираз може мати вигляд: "температура від'ємна", "температура, близька до нуля", "позитивна температура".

На рис. 3.5 наведена графічна ілюстрація належності змінної $x = T$ (де T означає температуру) для трьох названих множин значень температури.

ступінь приналежності

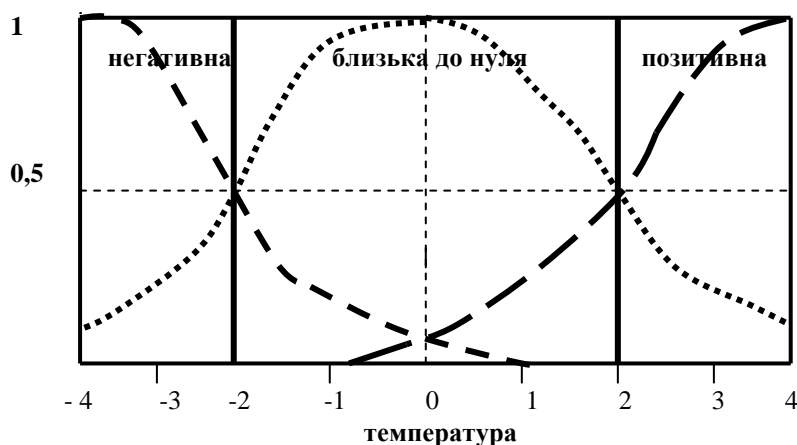


Рис.3.5. Ілюстрація поняття ступеня приналежності температури до області негативною, близькою до нуля або позитивною (пунктирні лінії – нечітка система, суцільні лінії – точна система)

3.2.1. Нечіткість і вірогідність

У теорії ймовірностей подія $u \in U$ (\in – знак належності елемента u множині U , про множину відомо, як мінімум, що вона

складається з елементів, що володіють деякою загальною властивістю) або відбувається, або ні, а ймовірність являє міру того, що вона відбудеться або випадкова змінна x прийме значення u . Оцінка ймовірності може бути розрахована як відношення кількості експериментів, в яких зазначена подія здійснилося, до загальної кількості експериментів. Наприклад, якщо день 21 листопада протягом останніх 100 років був дощовим 82 рази, то ймовірність дощу в цей день у цьому році оцінюється як 0,82. Слід підкреслити, що ймовірність події відноситься виключно до майбутнього. Коли відповідний день настане, дана подія або відбудеться, або ні, і в цей момент поняття ймовірності її звершення втрачає сенс. Наприклад, у момент початку дня 21 листопада все ще не ясно, чи буде цей день дощовим чи ні, однак, коли він завершиться, ми будемо говорити про те, чи сталася чи ні подія, а не про її ймовірність.

Поняття нечіткості оцінюється зовсім по-іншому. Воно вимірюється ступенем, з якою подія (наприклад, тільки що подія) належить до деякої множини подій A . Фактично вимірюється ступінь, в якій множина U міститься в домені A . Наприклад, якщо 21 листопада дощ йшов протягом 15 годин, то ступінь його приналежності до множини дощових днів можна визначити як $15/24$. З цієї точки зору поняття нечіткості відноситься не тільки до минулого, як це має місце у випадку ймовірності, але також до сьогодення та майбутнього.

Слід зазначити, що ймовірність може бути визначена як нечітке значення, особливо тоді, коли воно оцінюється приблизно, а не точним способом. Тому можна сказати, що ймовірність настання визначеної події становить, наприклад, "близько 0,7", оскільки змінна "близько 0,7" є лінгвістичною. Якщо ж нечітке поняття відноситься до майбутнього, йому можна приписати деяку ймовірність.

Приклад 3.3. Розглянемо ймовірність випадання кістки з певним номером з інтервалу $[1,6]$. Припустимо, що існують три

множини нечітких чисел: "мала", "середня" та "велика", функції належності до яких представлені на рис. 3.6а.

Ймовірність настання нечіткої події, що x - це мале, середнє або велике число, визначається за формулою:

$$p(x) = \sum_{i=1}^6 \mu(x_i) \cdot p(x_i),$$

де $p(x_i)$ – позначає ймовірність випадання певного числа з $[1,6]$.

Якщо допустити рівномірний розподіл ймовірності випадання кожного числа ($p(x_i) = 1/6$) так, як це представлено на рис. 3.10,б, отримаємо:

$$p(x = \text{"мала"}) = \frac{1}{6} \cdot (1 + 0,5) = 0,25,$$

$$p(x = \text{"середня"}) = \frac{1}{6} \cdot (0,5 + 1 + 0,5) = 0,333,$$

$$p(x = \text{"велика"}) = \frac{1}{6} \cdot (0,5 + 1 + 1) = 0,418.$$

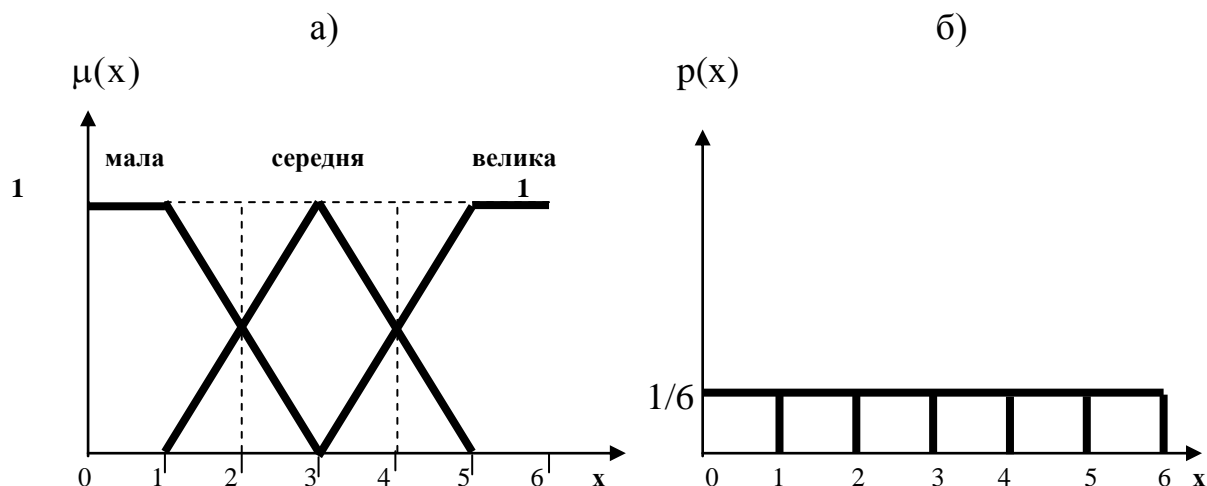


Рис. 3.6. Графічна ілюстрація даних прикладу викидання кістки: а) функції приналежності; б) ймовірність випадання відповідного номера

У наведеному результаті враховується поняття нечіткості лінгвістичної змінної: "мале число", "середнє число", "велика

кількість", а кожній події приписується ймовірність його настання.

3.2.2. Операції на нечітких множинах

На нечітких множинах, що розглядаються як узагальнення звичайних множин, можна визначити ряд математичних операцій, що є узагальненням аналогічних операцій, виконуваних на "чітких" множинах. Операції над нечіткими множинами дозволяють формувати інші нечіткі множини, відповідні складним явищам, подіям або властивостям, складеним з більш простих компонентів. Нехай є дві властивості A і B , кожна з яких може приймати значення з однієї множини $X = \{x_1, x_2, \dots, x_n\}$.

Група з N експертів оцінює можливість прийняття властивостям A і B значень $x_i (i = 1, 2, \dots, n)$. Нехай для параметра A значення x_i набрало N_i голосів, а для властивості B це ж значення x_i набрало M_i голосів. Тоді з цими властивостями пов'язані нечіткі множини

$$A = \{x_1 | p_1, \dots, x_n | p_n\}, \quad p_i = N_i / N;$$
$$B = \{x_1 | q_1, \dots, x_n | q_n\}, \quad q_i = M_i / N.$$

Операції над нечіткими множинами можна, взагалі кажучи, визначити різними способами. Вибір конкретно із них залежить від змісту, вкладеного у відповідні операції в рамках розглянутої задачі.

1. Об'єднання нечітких множин (логічна сума множин $A \cup B$).

Розглянемо властивість, яка полягає в тому, що має місце або властивість A , або властивість B , або обидві одночасно. Оцінимо число голосів R_i , яке набрало значення x_i для властивості C . Знаючи тільки числа N_i і M_i , ми не можемо точно сказати, яке значення R_i , однак неважко визначити, що

$$\max(N_i, M_i) \leq R_i \leq \min(N_i + M_i; N)$$

Нехай, наприклад, $N_i = 5, M_i = 8; N = 20$. Тоді $8 \leq R_i \leq 13$ (рис. 3.7).

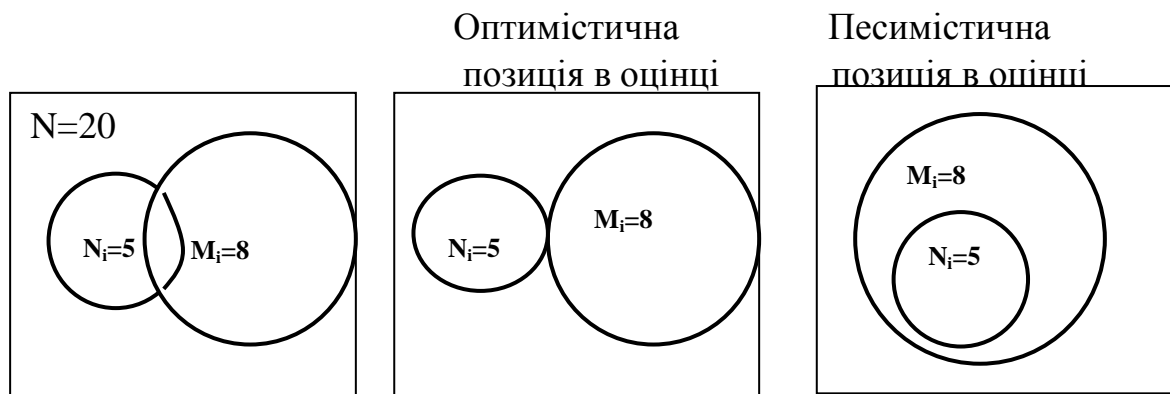


Рис. 3.7. Ілюстрація оцінки числа голосів

Якщо прийняти песимістичну позицію, відповідну крайній обережності в оцінці R_i , то мінімальне значення $R_i = \max(N_i, M_i)$, у нашому прикладі $R_i = 8$ (рис. 3.7, в).

Відповідно до прийнятого підходу, зв'яжемо з властивістю S нечітку множину

$$C = \{x_1 | r_1, \dots, x_n | r_n\}, \quad r_i = R_i / N.$$

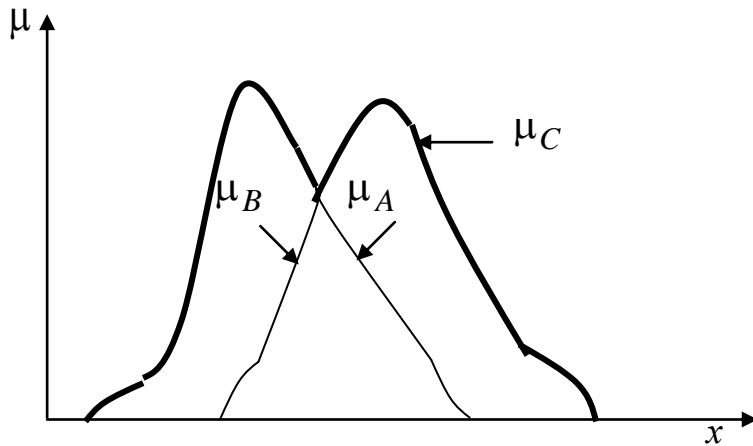
Таким чином, при позиції крайньої обережності в оцінці R_i

$$r_i = \max(p_i, q_i) \quad \text{або} \quad \mu_C(x_i) = \max(\mu_A(x_i), \mu_B(x_i)).$$

Визначення. Об'єднанням $A \cup B$ нечітких множин A і B однієї універсальної множини X , з відповідними функціями належності $\mu_A : X \rightarrow [0,1]$ и $\mu_B : X \rightarrow [0,1]$ називається нечітка множина C універсальної множини X , функція належності якої (рис.3.8) задається формулою:

$$\mu_C(x) = \max[\mu_A(x), \mu_B(x)].$$

Операція пов'язана з використанням союзу "АБО".

Рис. 3.8. Операція об'єднання двох нечітких множин A і B

Приклад 3.4. Нехай дано дві нечітких множини A і B , визначені таким чином:

$$A = \left\{ \frac{1,0}{x_1}, \frac{0,7}{x_2}, \frac{0,5}{x_3}, \frac{0,1}{x_4} \right\}, B = \left\{ \frac{0,2}{x_1}, \frac{0,5}{x_2}, \frac{0,6}{x_3}, \frac{0,7}{x_4} \right\}$$

Логічна сума цих множин $C = A \cup B$ дорівнює:

$$C = \left\{ \frac{1,0}{x_1}, \frac{0,7}{x_2}, \frac{0,6}{x_3}, \frac{0,7}{x_4} \right\}.$$

2. Перетин нечітких множин (логічний добуток множин $A \cap B$).

Розглянемо властивість D , яка полягає в тому, що має місце і властивість A , і властивість B . Оцінимо число голосів L_i , яке набрало значення x_i для властивості D . Нескладно визначити, що $M_i + N_i = R_i + L_i$.

Якщо орієнтуватися на принцип об'єднання нечітких множин, відповідною формулою $\mu_C(x) = \max[\mu_A(x), \mu_B(x)]$ та рис. 3.8, то для визначення L_i слід прийняти $L_i = \min(N_i, M_i)$. Звідси, вважаючи $l_i = L_i / N$, отримаємо $l_i = \min(p_i, q_i)$ або $\mu_D(x_i) = \min[\mu_A(x_i), \mu_B(x_i)]$.

Визначення. Перетином $A \cap B$ нечітких множин A і B універсальної множини X з функціями належності відповідно $\mu_A : X \rightarrow [0,1]$ і $\mu_B : X \rightarrow [0,1]$ називається нечітка множина D універсальної множини X , функція належності якої (рис. 3.9) задається формулою

$$\mu_D(x_i) = \min[\mu_A(x_i), \mu_B(x_i)].$$

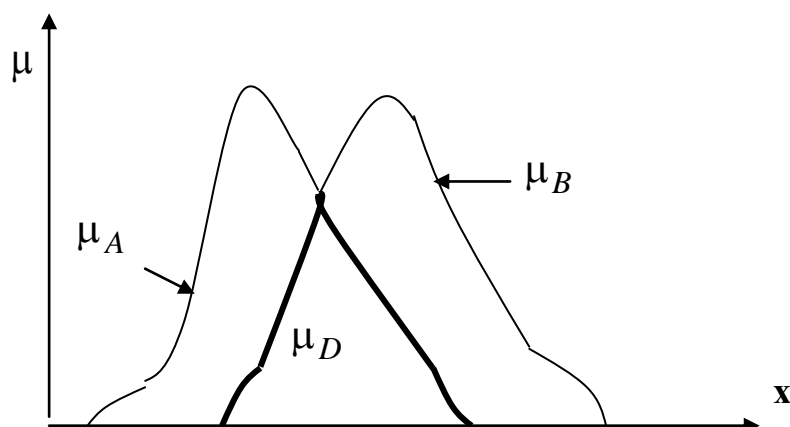


Рис. 3.9. Операція перетину двох нечітких множин A і B

Зазначимо, що операція перетину пов'язана з використанням союзу "І".

Для даних з прикладу 3.3 множину, яка є логічним добутком множин A і B , буде мати вигляд

$$C = \left\{ \frac{0,2}{x_1}, \frac{0,5}{x_2}, \frac{0,5}{x_3}, \frac{0,1}{x_4} \right\}.$$

3. Додаток нечітких множин (заперечення множини \bar{A})

Поняття доповнення \bar{A} нечіткої множини A пов'язується з властивістю E , полягає в тому, що властивість A не має сенсу. Якщо властивість A для значення x_i набрала N_i голосів, то властивість E – відповідно $(N - N_i)$ голосів. Зв'яжемо з властивістю E нечітку множину

$$E = \{x_1 | k_1, \dots, x_n | k_n\}, \quad k_i = (N - N_i) / N = 1 - p_i.$$

Звідки

$$\mu_{\bar{A}}(x_i) = 1 - \mu_A(x_i).$$

Визначення. Доповненням \bar{A} нечіткої множини A універсальної множини X називається множина E з функцією належності

$$\mu_{\bar{A}}(x_i) = 1 - \mu_A(x_i)$$

Операція доповнення пов'язана з використанням заперечення "НІ".

На рис. 3.10 графічно показано перетин нечітких множин $A \cap \bar{A}$. Рівень відмінності A й \bar{A} визначає міру нечіткості множини A .

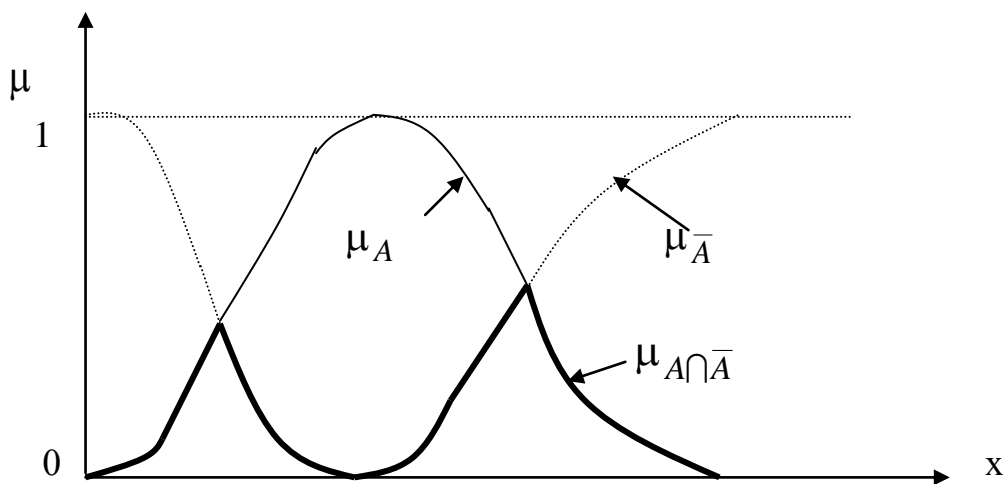


Рис. 3.10. Перетин нечітких множин $A \cap \bar{A}$

На відміну від звичайних множин (чітких) множин, де заперечення елементів, що належать до множини, дає порожня множина, заперечення нечіткої множини визначає непорожня множина, що складається з елементів, функції належності яких також визначені на інтервалі $[0, 1]$.

Приклад 3.5. Нехай універсальна множина $U = 1 + 2 + 3 + 4$, а нечітку множину A , що характеризує поняття "високий", задано у вигляді

$$A = \text{високий} = 0/1 + 0,6/2 + 0,8/3 + 1/4.$$

Тоді

$$\bar{A} = \text{не високий} = 1/1 + 0,4/2 + 0,2/3 + 0/4.$$

4. Рівність множин A і B

Нечіткі множини та рівні між собою, коли для всіх елементів обох множин виконується умова $\mu_A(x_i) = \mu_B(x_i)$.

5. Операція концентрації $CON(A)$

$$\mu_{CON}(x) = [\mu_A(x)]^2.$$

Виконуючи операцію концентрації для прикладу 3.4, одержимо:

$$B = CON(A) = \text{дуже високий} = 0/1 + 0,36/2 + 0,64/3 + 1/4.$$

Ця операція досить часто виконується при діях з лінгвістичною змінною, в яких вона ототожнюється з інтенсифікатором "дуже".

6. Операція розтягування.

$$\mu_{DIL}(x) = [\mu_A(x)]^{0,5}$$

Лінгвістичне значення цієї операції формулюється як "приблизно" або приблизно. Ця операція протилежна за своїм змістом операції концентрації і відповідає терму «досить». Операція розтягування використовується у випадках, коли потрібно моделювати втрату інформації.

7. Алгебраїчний добуток двох множин $A \cdot B$

$$\mu_{A \cdot B}(x) = \mu_A(x) \cdot \mu_B(x)$$

Алгебраїчний добуток для прикладу 3.3 буде дорівнювати:

$$C = \left\{ \frac{1}{x_1}, \frac{0,49}{x_2}, \frac{0,3}{x_3}, \frac{0,07}{x_4} \right\}$$

8. Обмежена сума двох нечітких множин $A|+|B$

$$\mu_{A|+|B}(x) = \min \{1, \mu_A(x) + \mu_B(x)\}.$$

Обмежена сума двох нечітких множин для прикладу 3.3 буде дорівнювати:

$$C = \left\{ \frac{0,8}{x_1}, \frac{0,2}{x_2}, \frac{0}{x_3}, \frac{0}{x_4} \right\}$$

9. Обмежена різниця двох нечітких множин $A|-|B$

$$\mu_{A|-|B}(x) = \max\{0, \mu_A(x) - \mu_B(x)\}.$$

Обмежена різниця двох нечітких множин для прикладу 3.3 буде дорівнювати:

$$C = \left\{ \frac{0,2}{x_1}, \frac{0,2}{x_2}, \frac{0,1}{x_3}, \frac{0}{x_4} \right\}$$

10. Обмежений добуток двох нечітких множин $A|\cdot|B$

$$\mu_{A|\cdot|B}(x) = \max\{0, \mu_A(x) + \mu_B(x) - 1\}$$

Обмежений добуток двох нечітких множин для прикладу 3.3 буде дорівнювати:

$$C = \left\{ \frac{0,2}{x_1}, \frac{0,2}{x_2}, \frac{0,1}{x_3}, \frac{0}{x_4} \right\}$$

11. Нормалізація множини $NORM(A)$

$$\mu_{NORM}(x) = \frac{\mu_A(x)}{\max\{\mu_A(x)\}}$$

Зазначимо, що множина A вважається підмножиною множини B , тобто $A \subset B$, коли для всіх елементів виконується нерівність $\mu_A(x_i) \leq \mu_B(x_i)$. Наприклад, якщо $A = \{0,5/x_1, 0,3/x_2, 0,1/x_3\}$ і $B = \{0,6/x_1, 0,5/x_2, 0,4/x_3\}$ то $A \subset B$.

Визначені на нечітких множинах операції мають властивості асоціативності, комутативності та дистрибутивності, причому ці властивості розуміються таким чином:

- асоціативність: $(A * B) * C = A * (B * C)$;
- комутативність: $A * B = B * A$ (за винятком обмеженої різниці);
- дистрибутивність: $A * (B \circ C) = (A * B) \circ (A * C)$,

де оператори $*$ і \circ позначають будь-яку наведену вище операцію на нечітких множинах.

3.2.3. Основні типи функцій приналежності

Формальне визначення нечіткої множини не накладає ніяких обмежень на вибір конкретної функції приналежності для його подання. Однак на практиці зручно використовувати ті з них, які допускають аналітичне представлення у вигляді деякої простої математичної функції. Це спрощує не тільки відповідні чисельні розрахунки, але і скорочує обчислювальні ресурси, необхідні для зберігання окремих значень цих функцій приналежності. Необхідність типізації окремих функцій належності також обумовлена наявністю реалізацій відповідних функцій у розглянутих далі інструментальних засобах.

Кусково-лінійні функції належності

В якості першого типу функцій приналежності розглянемо функції, які, як випливає з їх назви, складаються з відрізків прямих ліній, утворюючи безперервну або кусковонеперервну функцію.

Найбільш характерним прикладом таких функцій є "трикутна" (рис. 3.11,а) і "трапецієвидна" (рис.3.11,б) функції приналежності.

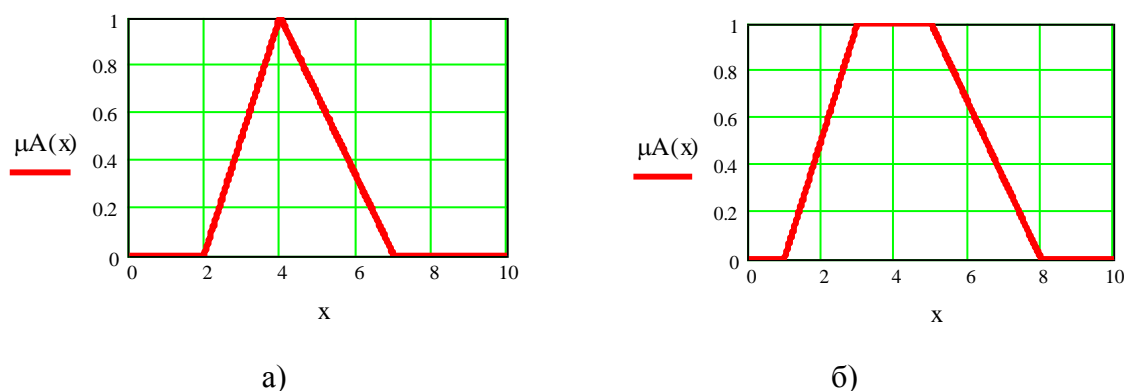


Рис.3.11. Графіки функцій приналежності трикутної (а) і трапецієподібної (б) форми

В нашому випадку кожна з цих функцій задана на універсумі $X=[0, 10]$, в якості якого обрано замкнутий інтервал дійсних чисел. У загальному випадку вибір універсуму може бути довільним і не обмежений ніякими правилами.

Перша з цих функцій належності в загальному випадку може бути задана аналітично таким виразом:

$$f_{\Delta}(x, a, b, c) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ \frac{c-x}{c-b}, & b \leq x \leq c \\ 0, & c \leq x \end{cases}$$

де a, b, c – деякі числові параметри, що приймають довільні дійсні значення і впорядковані співвідношенням $a \leq b \leq c$ (у нашому прикладі $a=2, b=4, c=7$).

Стосовно до конкретної функції, зображеної на рис. 3,11, а значення параметрів дорівнюють: $a=2, b=4, c=7$. Як неважко помітити, параметри a і c характеризують підставу трикутника, а параметр b його вершину. Як можна помітити, ця функція приналежності породжує нормальну опуклу унімодальну нечітку множину з носієм інтервалом (a, c) , кордонами $(a, c) \setminus \{b\}$, ядром $\{b\}$ і модою b .

Трапецеподібна функція приналежності в загальному випадку може бути задана аналітично таким виразом:

$$f_T(x, a, b, c, d) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ 1, & b \leq x \leq c \\ \frac{d-x}{d-c}, & c \leq x \leq d \\ 0, & d \leq x \end{cases},$$

де a, b, c, d – деякі числові параметри, що приймають довільні дійсні значення і в порядкуванні співвідношенням $a \leq b \leq c \leq d$ (у нашому прикладі $a=1, b=3, c=5, d=8$).

Ці функції використовуються для завдання таких властивостей множин, які характеризують невизначеність типу: "приблизно дорівнює", "середнє значення", "розташований в інтервалі", "подібний до об'єкту", "схожий на предмет" та ін. Вони також служать для представлення нечітких чисел і інтервалів.

Z-образні і S-образні функції належності

Ці функції приналежності також отримали свою назву по виду кривих, які представляють їх графіки. До типу S-подібних і одночасно Z-подібних функцій належності може бути віднесена так звана *сигмоїдальна* функція (сигмоїд), яка в загальному випадку задається аналітично таким виразом:

$$f_S(x, a, b) = \frac{1}{1 + \exp[-a \cdot (x - b)]},$$

де a, b – деякі числові параметри, що приймають довільні дійсні значення і в порядкуванні співвідношенням $a \leq b$, а \exp – основа натуральних логарифмів, яка ініціює завдання відповідної експоненційної функції.

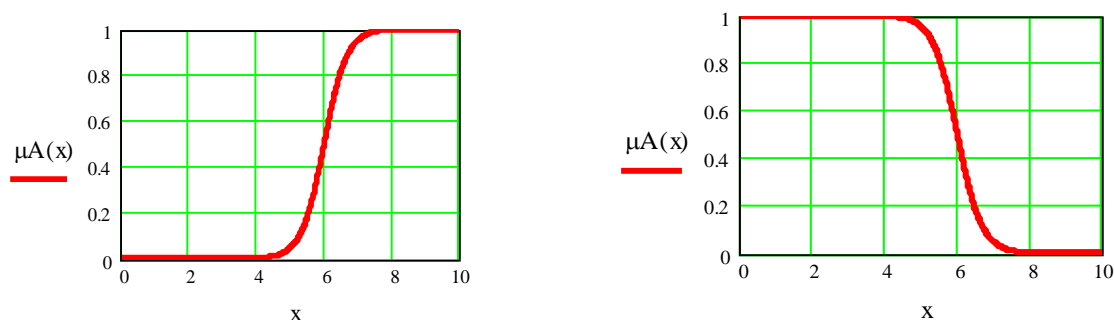


Рис. 3.12. Графіки сигмоїдної функції приналежності для значень параметрів $a=3, b=6$ (а) і $(a=-3, b=6)$ (б) (b – точка переходу)

При цьому в разі $a > 0$ може бути отримана S -подібна функція, а в разі $a < 0$ – Z -подібна функція приналежності. Графіки цієї функції для деякої нечіткої множини A і універсуму $X = [0,10]$ представлені на рис.3.12.

S -подібні функції приналежності використовуються для подання таких нечітких множин, які характеризуються невизначеністю типу: "велика кількість", "велике значення", "значна величина", "високий рівень доходів і цін", "висока норма прибутку", "висока якість послуг" та багато інших. Загальним для всіх таких ситуацій є висока ступінь прояву тієї чи іншої якісної або кількісної ознаки. Особливість нечіткого моделювання при цьому полягає в поданні відповідних нечітких множин за допомогою неспадних (монотонно зростаючих) функцій приналежності.

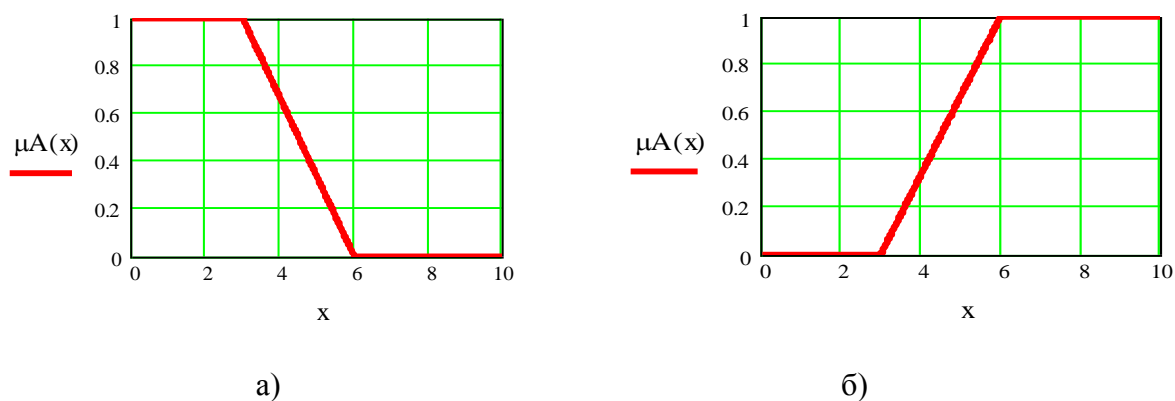


Рис. 3.13. Графіки лінійної Z -подібної функції (а) і лінійної S -подібної функції(б) належності для значень параметрів $a=3$, $b=6$

В якості приватних випадків і образних кривих зручно розглядати так звану лінійну образну функцію (рис. 3.13, а) і лінійну образну функцію (рис.3.13, б). Перша з цих функцій у загальному випадку може бути задана аналітично таким виразом:

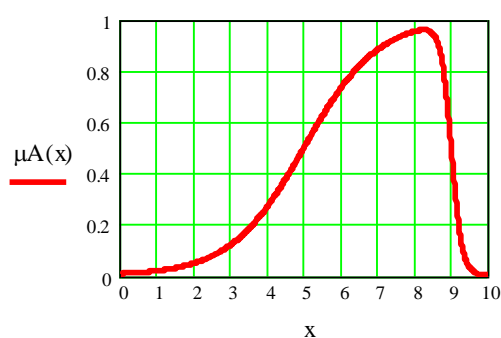
$$f_{\downarrow}(x, a, b) = \begin{cases} 1, & x \leq a \\ \frac{b-x}{b-a}, & a < x < b \\ 0, & b \leq x \end{cases}$$

де a, b – деякі числові параметри, що приймають довільні дійсні значення і впорядковані співвідношенням

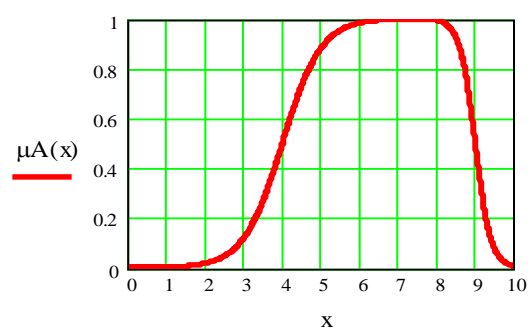
Друга з цих функцій у загальному випадку може бути задана аналітично таким виразом:

$$f_{\uparrow}(x, a, b) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a < x < b \\ 1, & b \leq x \end{cases}$$

П-подібні функції приналежності. До даного типу функцій приналежності можна віднести цілий клас кривих, які за своєю формою нагадують дзвін, згладжену трапецію або букву "П".



а)



б)

Рис. 3.14. Графіки П-подібних функцій приналежності f_{Π} для значень параметрів $a=1, b=5, c=-7, d=9$ (а) і для значень $a=2, b=4, c=-5, d=9$

Функція цього типу може визначатися як добуток двох сигмоїдальних функцій і в загальному випадку задається аналітичним виразом:

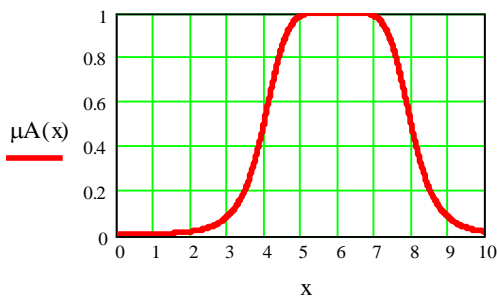
$$f_{\Pi}(x, a, b, c, d) = f_S(x, a, b) \cdot f_S(x, c, d),$$

де a, b, c, d – деякі числові параметри, що приймають довільні дійсні значення, причому $a > 0, c < 0$, і впорядковані відношенням $a \leq b \leq |c| \leq d$ (рис.3.14).

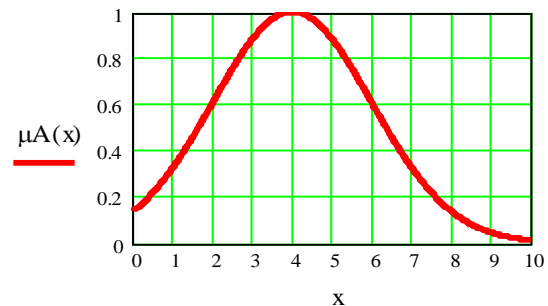
До П-подібних функцій відноситься також так звана дзвоноподібна функція, яка у загальному випадку задається аналітично таким виразом:

$$f_{\Pi}(x, a, b, c) = \frac{1}{1 + \left| \frac{x - c}{a} \right|^{2b}},$$

де a, b, c – деякі числові параметри, що приймають довільні дійсні значення і впорядковані співвідношенням $a \leq b \leq c$, причому параметр $b > 0$ (рис.3.15, а).



а)



б)

Рис. 3.15. Графіки П-подібних функцій приналежності для значень $a = 2, b = 3, c = 6$ (а) і функція щільності нормального розподілу для значень параметрів $\sigma = 2, c = 4$ (б)

І, нарешті, остання з розглянутих функцій даного типу є добре відомою в теорії ймовірностей функція щільності

нормального розподілу (рис.3.15, б) у припущенні, що $\sqrt{2\pi}\sigma = 1$, і яка задається таким виразом:

$$f_{\Pi}(x, \sigma, c) = \exp\left[\frac{-(x-c)^2}{2\sigma^2}\right]$$

де σ, c – числові параметри, при цьому квадрат першого з них називається в теорії ймовірностей *дисперсією*, а другий параметр c – *математичним очікуванням*.

Загальна форма гауссівської функції для змінної x з центром з і варіацією σ визначається формулою:

$$\mu_A(x) = \exp\left[-\left(\frac{x-c}{\sigma}\right)^2\right].$$

На рис. 3.16 представлена форма типових гауссівських функцій при різних значеннях параметрів c та σ , причому на рис. 3.16, а показано вплив розміщення центру c при незмінному значенні σ , а на рис.3.16, б – вплив значення σ при фіксованому c .

Параметр c позначає центр нечіткої множини, а його зміна відповідає зсуву функції приналежності по горизонтальній осі. Параметр σ , іноді званий коефіцієнтом широти, відповідає за форму функції. Чим менше його значення, тим більше крутизна функції. Слід зазначити, що при відповідному зміщенні центру Гауссова функція може реалізувати і сигмоїдальну функцію (найчастіше при зсуві вправо з $c = 4$).

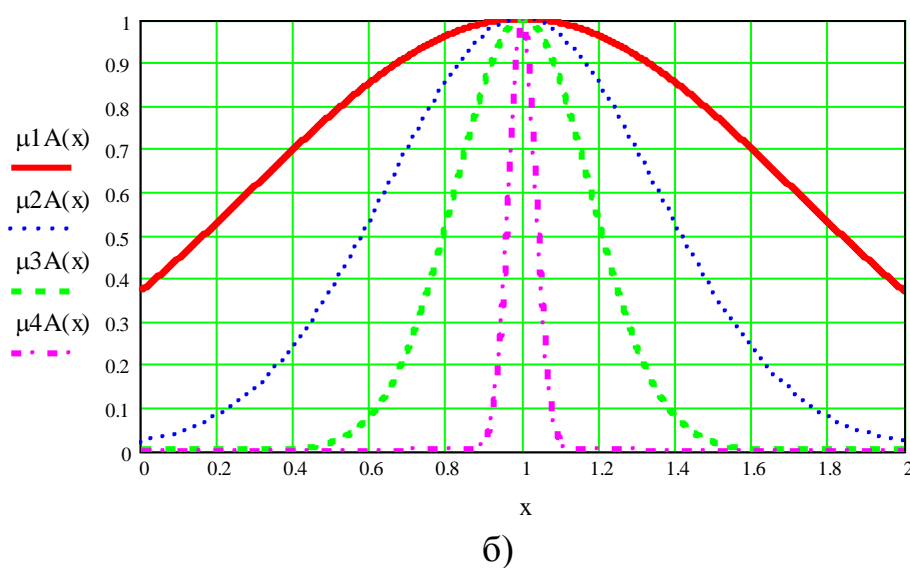
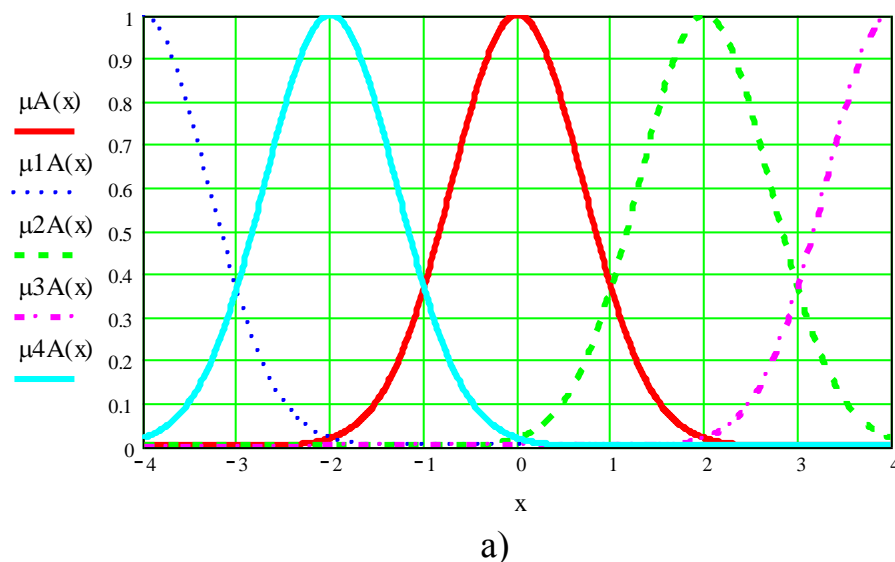
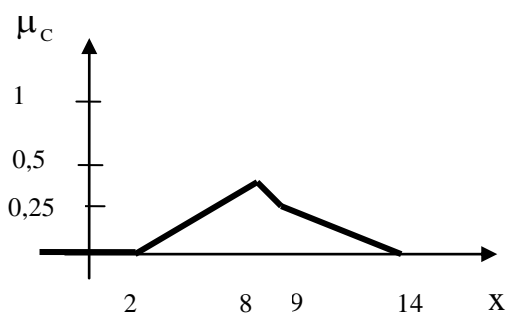
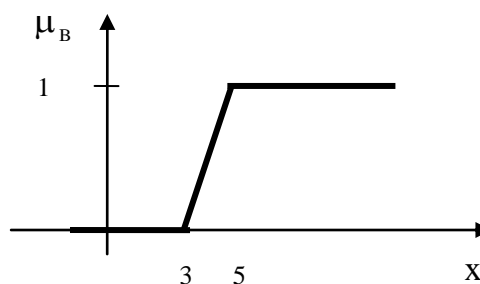
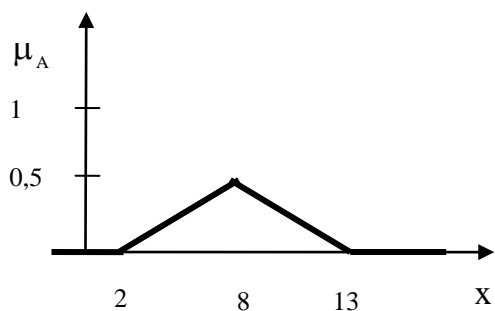


Рис. 3.16. Ілюстрація впливу параметрів гауссівської функції на її форму: а) вплив розміщення центру c при $\sigma = 1$, $c = -4$, $c = -2$, $c = 0$, $c = 2$, $c = 4$; б) вплив значення σ при постійному $c = 1$, $\sigma = 1$, $\sigma = 0,5$, $\sigma = 0,25$, $\sigma = 0,05$

Приклад 3.6. Завдання. Дано 3 нечітких множини A , B , C (задані їх функції приналежності). Побудувати функцію приналежності нечіткої множини $D = \bar{A} \cap (A \cup C \cup B)$ і визначити

ступінь належності одного елемента множині D , використовуючи метод обмежень.

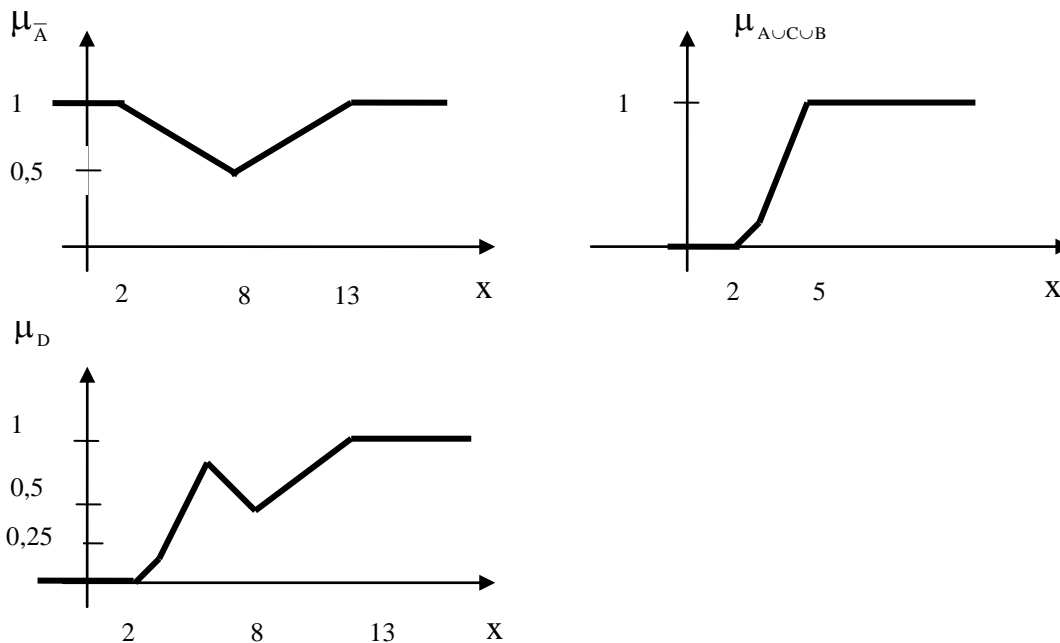


Опис процесу рішення. Для побудови функції приналежності нової множини необхідно:

- 1) Визначити послідовність виконання операцій у формулі.
- 2) Побудувати на окремих графіках проміжні множини, згідно з визначеною послідовністю дій. Звести проміжні множини на одному графіку та визначити підсумкову функцію приналежності.
- 3) Використовуючи визначений у завданні метод, визначити аналітично ступінь приналежності елемента, що входить до ядра підсумкової множини.
- 4) Перевірити аналітичні обчислення за побудованим графіком функції приналежності.

Розв'язання.

- 1) Множина $D = \bar{A} \cap (A \cup C \cup B)$, значить, послідовність операцій буде такою: $\bar{A}, A \cup C \cup B, \bar{A} \cap (A \cup C \cup B)$.
- 2) Побудуємо відповідно до цієї послідовності операцій графіки функцій приналежності:



3) Ядро множини D складається з елементів з інтервалу (2,13).
Оберемо елемент 8.

$$\mu_A(8) = 0,5;$$

$$\mu_B(8) = 1;$$

$$\mu_C(8) = 0,5;$$

$$\mu_{\bar{A}}(8) = 1 - \mu_A(8) = 1 - 0,5 = 0,5;$$

$$\mu_{\bar{C}}(8) = 1 - \mu_C(8) = 1 - 0,5 = 0,5;$$

$$\mu_{A \cup C}(8) = \min\{1, \mu_C(8) + \mu_A(8)\} = \min\{1, 0,5 + 0,5\} = 1;$$

$$\mu_{A \cup C \cup B}(8) = \min\{1, \mu_{A \cup C}(8) + \mu_B(8)\} = \min\{1, 1 + 1\} = 1;$$

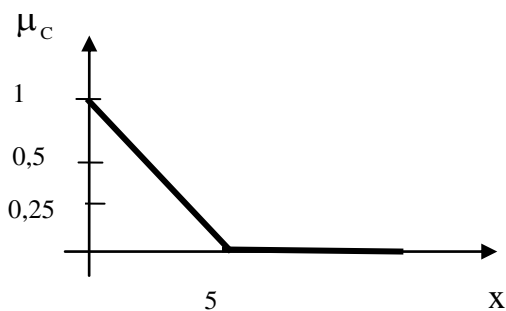
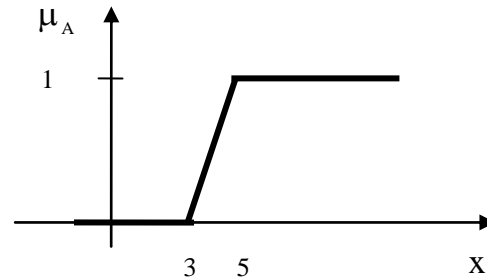
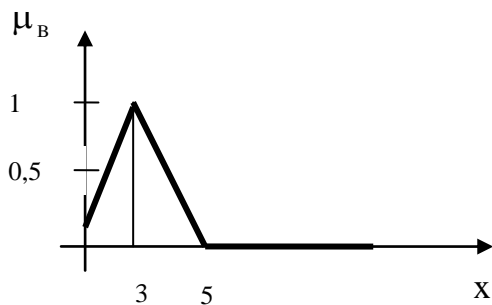
$$\mu_{\bar{A} \cap (A \cup C \cup B)}(8) = \max\{0, \mu_{\bar{A}}(8) + \mu_{A \cup C \cup B}(8) - 1\} = \max\{0, 0,5 + 1 - 1\} = 0,5;$$

$$\mu_D(8) = 0,5$$

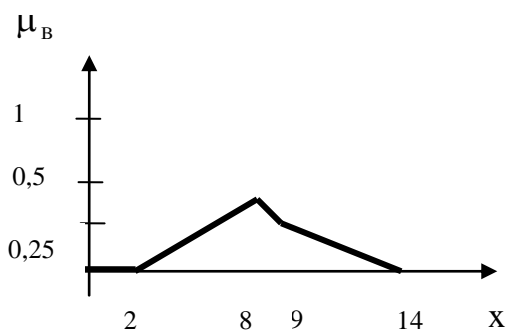
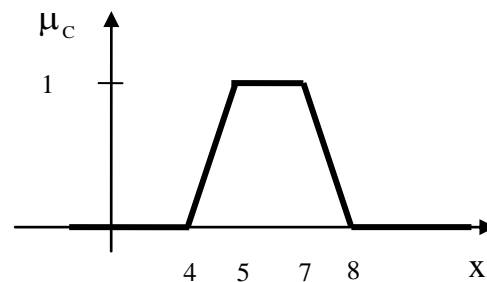
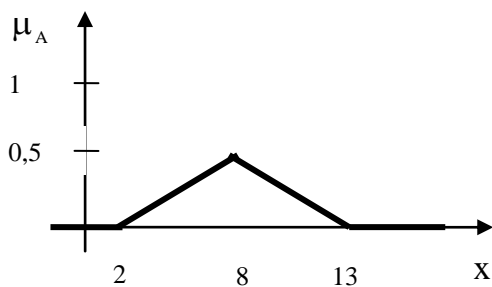
Задачі для самостійного розв'язання

1) Дано 3 нечітких множини A, B, C (задані їх функції приналежності). Побудувати функцію приналежності нечіткої множини $D = A \cup B \cap C$ і визначити ступінь приналежності одного елемента множини D, використовуючи максимінний спосіб.

2)

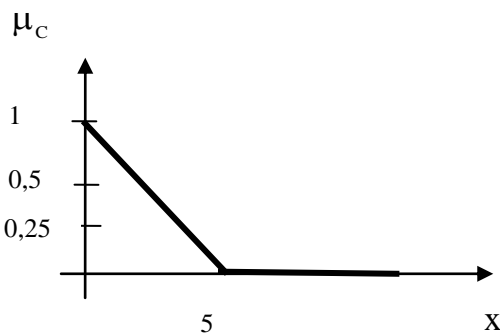
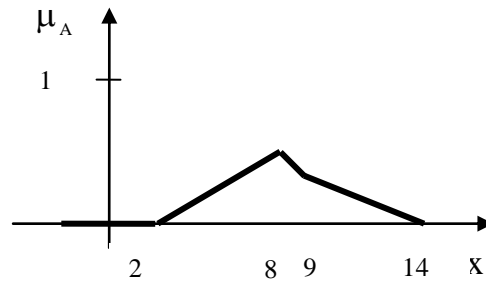
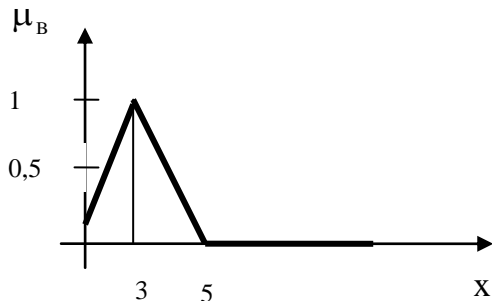


3) Дано 3 нечітких множини A , B , C (задані їх функції приналежності). Побудувати функцію приналежності нечіткої множини $D = A \cup B \cap C$ і визначити ступінь приналежності одного елемента множини D , використовуючи алгебраїчний спосіб.

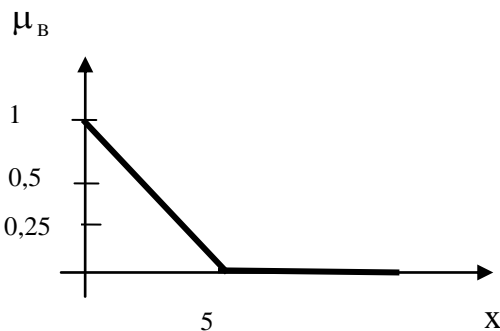
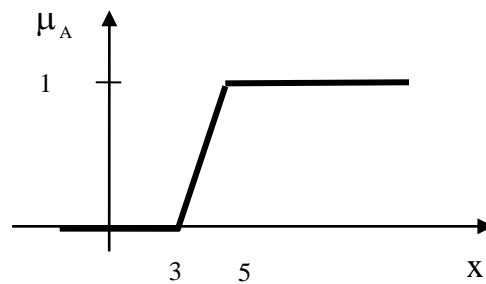
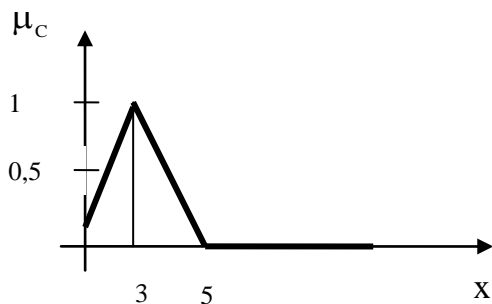


4) Дано 3 нечітких множини A , B , C (задані їх функції приналежності). Побудувати функцію приналежності нечіткої

множини $D = A \cup B \cap C$ визначити ступінь приналежності одного елемента множини D , використовуючи метод обмежень.

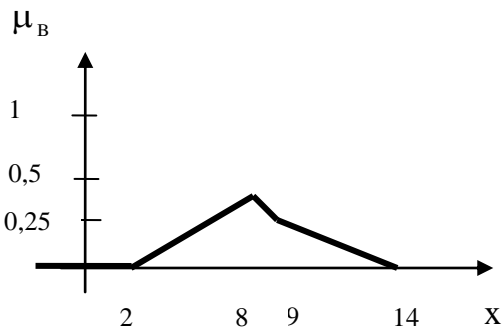
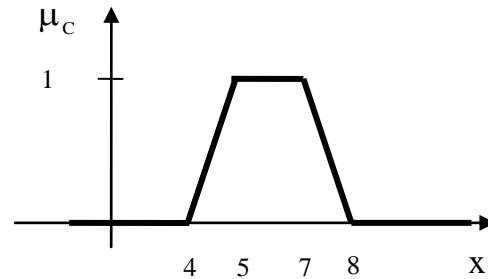
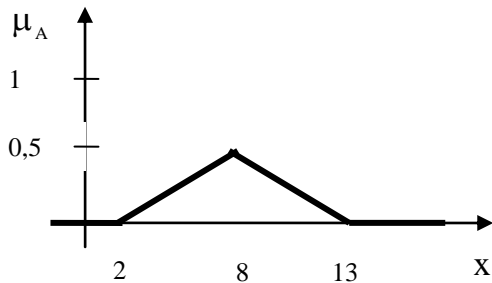


5) Дано 3 нечітких множини A , B , C (задані їх функції приналежності). Побудувати функцію приналежності нечіткої множини $D = A \cup \bar{B} \cap C$ і визначити ступінь приналежності одного елемента множини D , використовуючи максимінний спосіб.

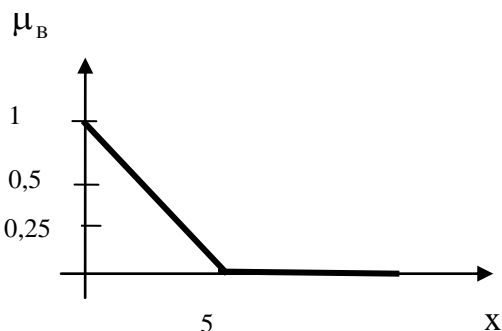
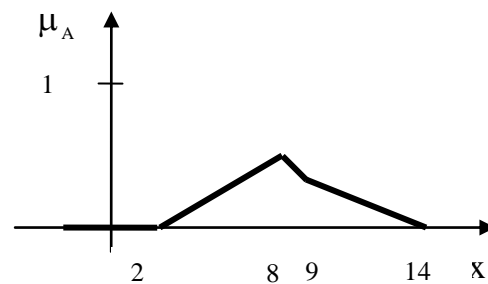
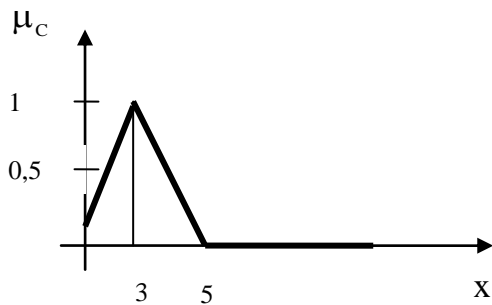


5) Дано 3 нечітких множини A , B , C (задані їх функції приналежності). Побудувати функцію приналежності нечіткої

множини $D = A \cup \bar{B} \cap C$ і визначити ступінь приналежності одного елемента множини D , використовуючи алгебраїчний спосіб.

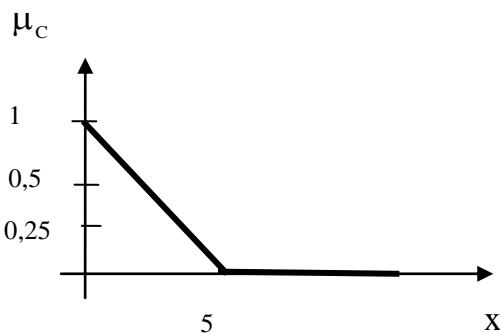
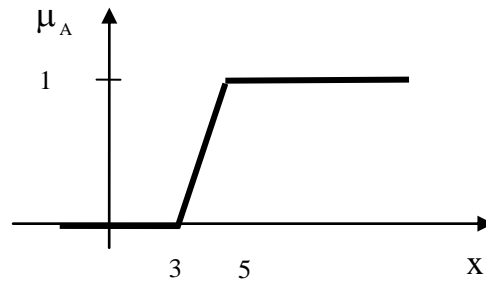
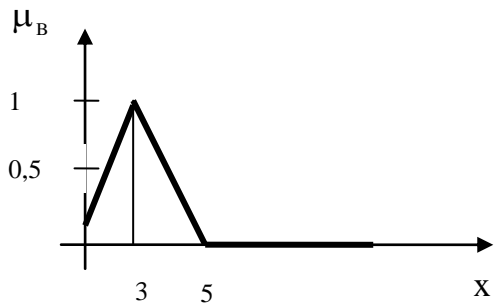


б) Дано 3 нечітких множини A , B , C (задані їх функції приналежності). Побудувати функцію приналежності нечіткої множини $D = A \cup \bar{B} \cap C$ і визначити ступінь приналежності одного елемента множини D , використовуючи метод обмежень.

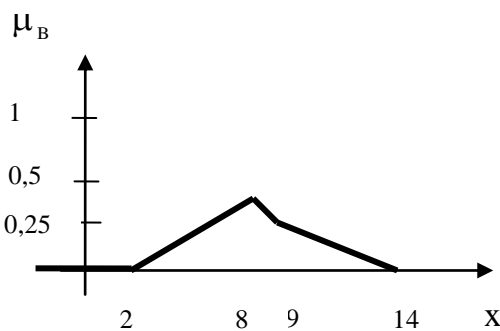
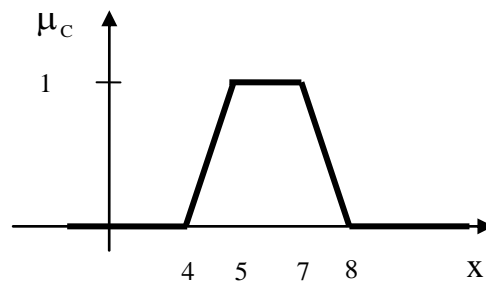
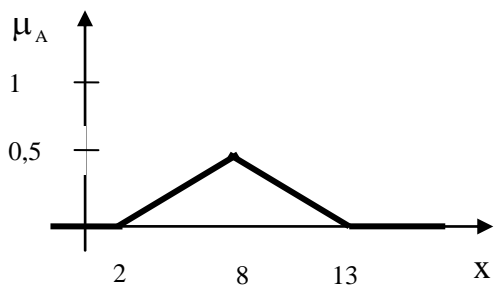


7) Дано 3 нечітких множини A , B , C (задані їх функції приналежності). Побудувати функцію приналежності нечіткої

множини $D = A \cap B \cup \bar{C}$ і визначити ступінь приналежності одного елемента множини D , використовуючи максимінний спосіб.

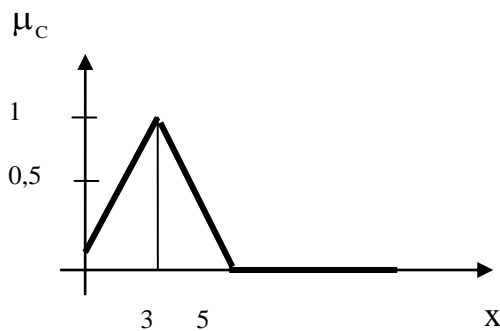
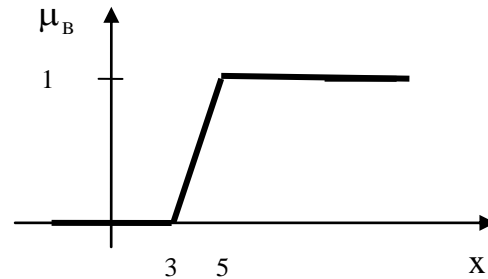
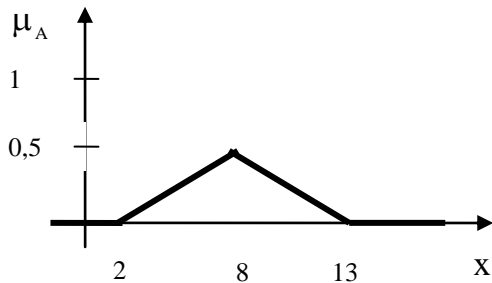


8) Дано 3 нечітких множини A , B , C (задані їх функції приналежності). Побудувати функцію приналежності нечіткої множини $D = A \cap B \cup \bar{C}$ і визначити ступінь приналежності одного елемента множини D , використовуючи алгебраїчний спосіб.

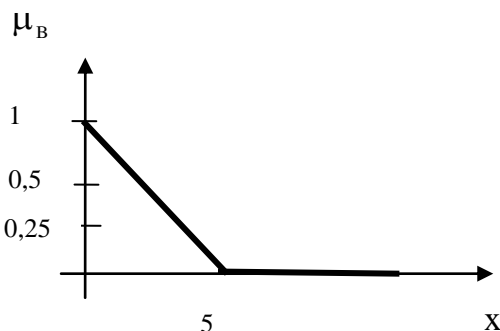
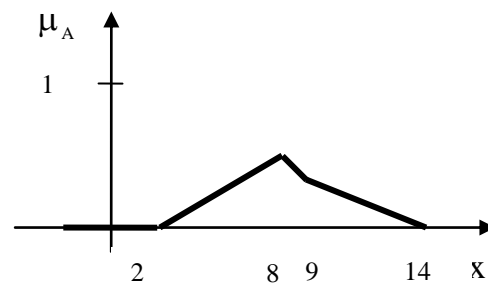
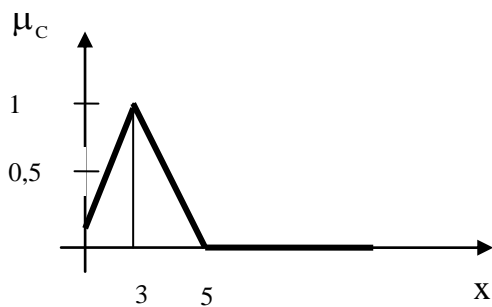


9) Дано 3 нечітких множини A , B , C (задані їх функції приналежності). Побудувати функцію приналежності нечіткої

множини $D = A \cap \bar{B} \cup \bar{C}$ і визначити ступінь приналежності одного елемента множини D , використовуючи метод обмежень.

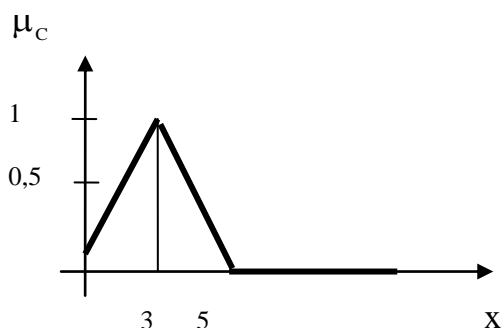
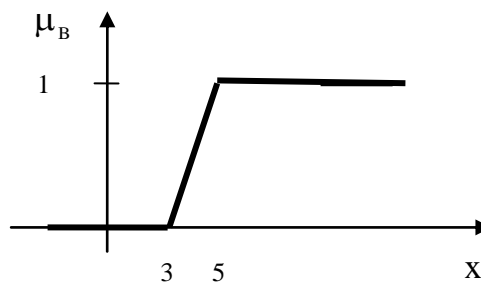
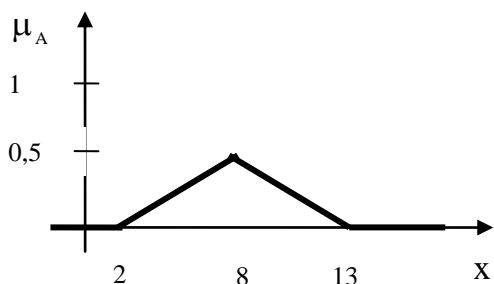


10) Дано 3 нечітких множини A , B , C (задані їх функції приналежності). Побудувати функцію приналежності нечіткої множини $D = \bar{A} \cup B \cap C$ і визначити ступінь приналежності одного елемента множини D , використовуючи максимінний спосіб.

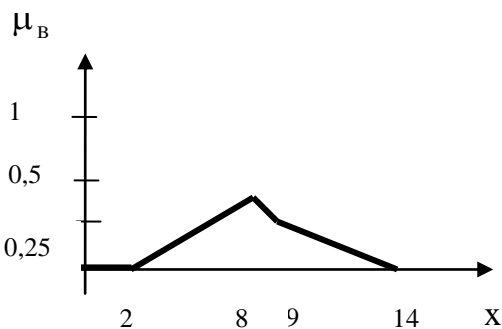
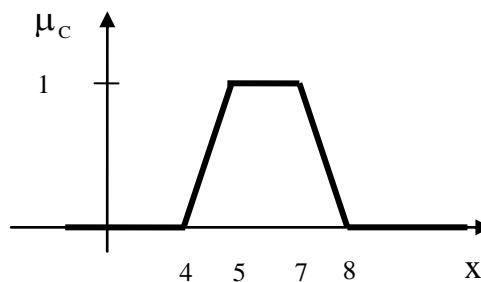
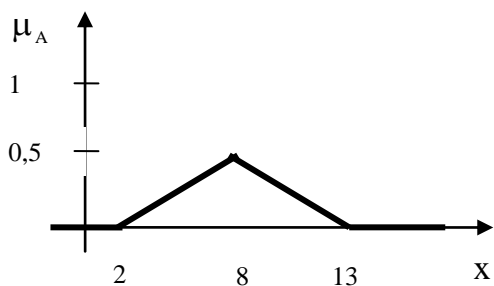


11) Дано 3 нечітких множини A , B , C (задані їх функції приналежності). Побудувати функцію приналежності нечіткої

множини $D = \bar{A} \cup B \cap C$ і визначити ступінь приналежності одного елемента множини D , використовуючи алгебраїчний спосіб.

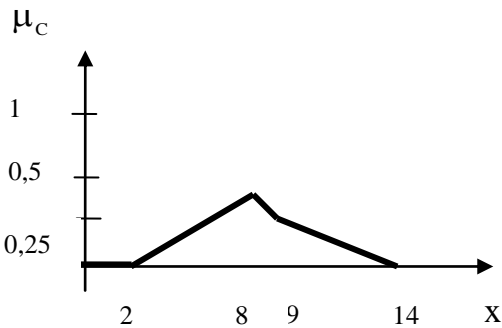
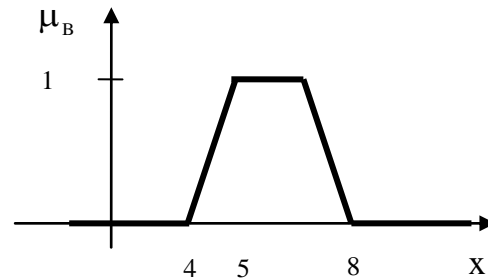
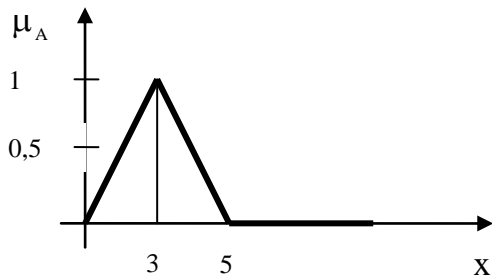


12) Дано 3 нечітких множини A , B , C (задані їх функції приналежності). Побудувати функцію приналежності нечіткої множини $D = \bar{A} \cup B \cap C$ і визначити ступінь приналежності одного елемента множини D , використовуючи метод обмежень.

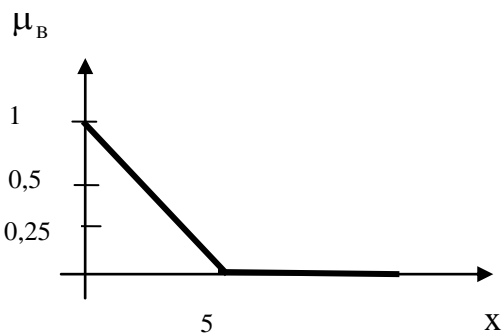
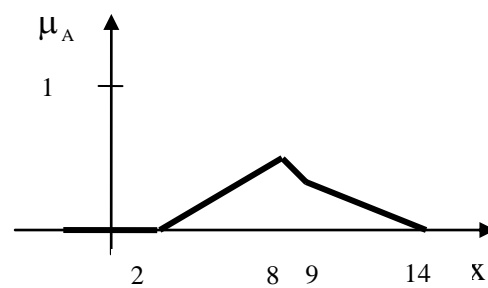
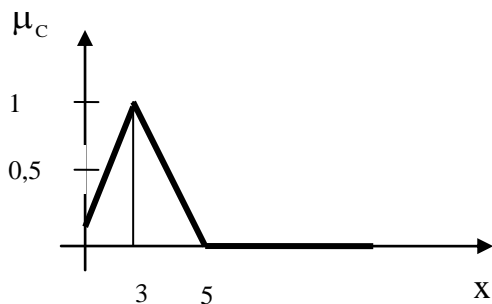


13) Дано 3 нечітких множини A , B , C (задані їх функції приналежності). Побудувати функцію приналежності нечіткої

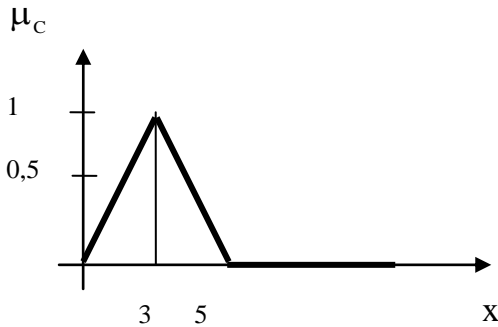
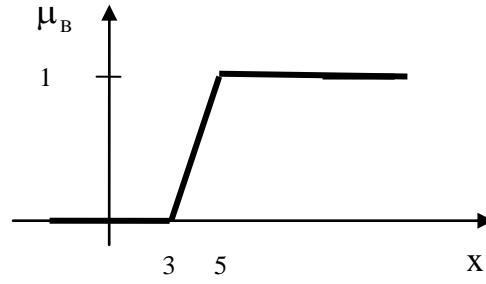
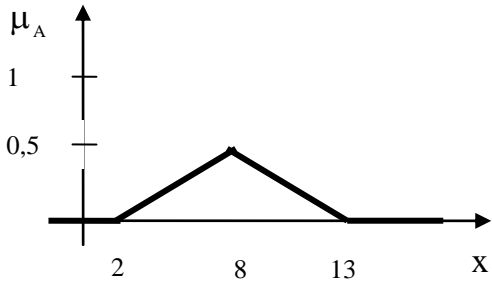
множини $D = (\bar{A} \cup B) \cap \bar{C}$ і визначити ступінь приналежності одного елемента множини D , використовуючи максимінний спосіб.



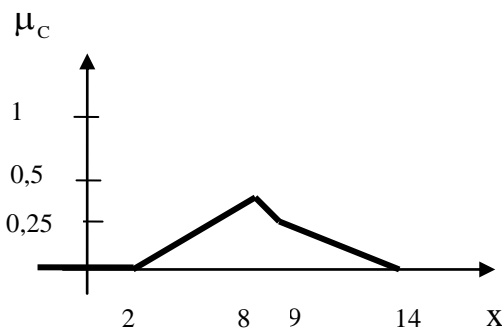
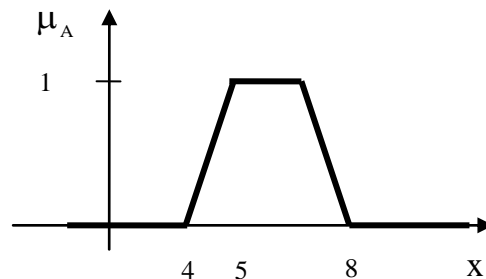
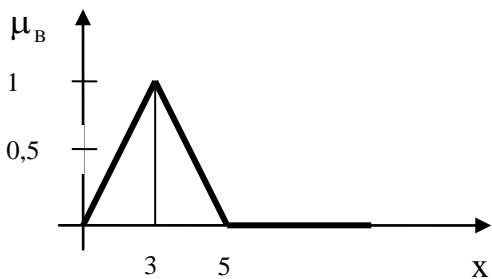
14) Дано 3 нечітких множини A , B , C (задані їх функції приналежності). Побудувати функцію приналежності нечіткої множини $D = (\bar{A} \cup B) \cap \bar{C}$ і визначити ступінь приналежності одного елемента множини D , використовуючи алгебраїчний спосіб.



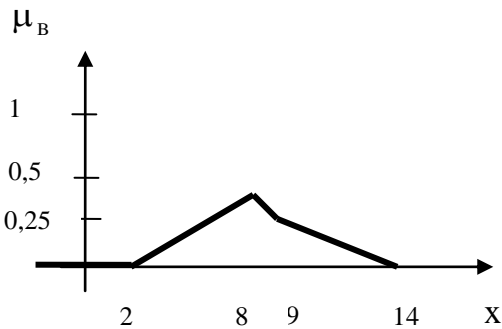
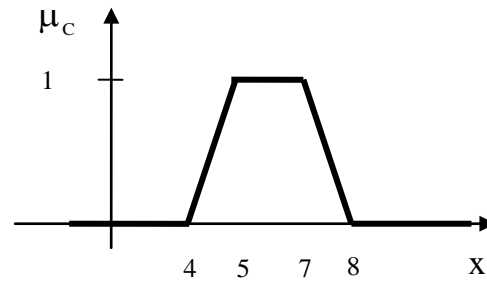
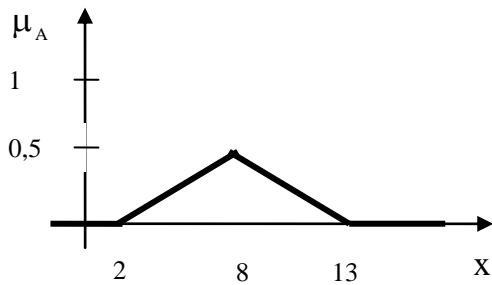
15) Дано 3 нечітких множини A, B, C (задані їх функції приналежності). Побудувати функцію приналежності нечіткої множини $D = (\bar{A} \cup B) \cap \bar{C}$ і визначити ступінь приналежності одного елемента множини D , використовуючи спосіб обмежень.



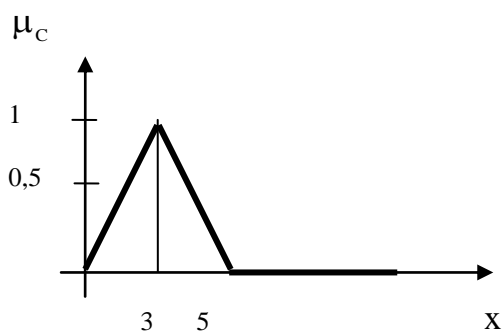
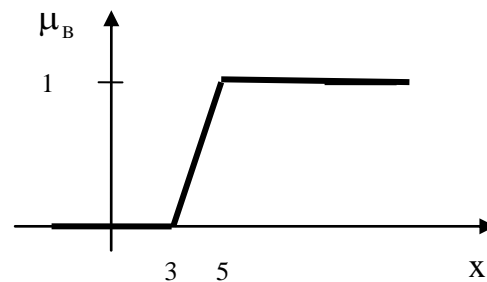
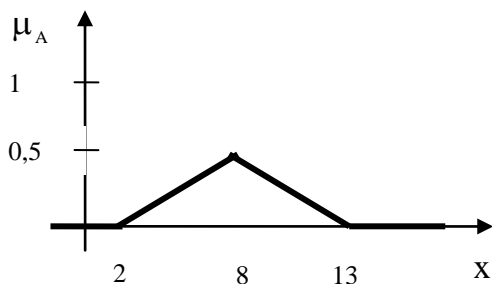
16) Дано 3 нечітких множини A, B, C (задані їх функції приналежності). Побудувати функцію приналежності нечіткої множини $D = \bar{A} \cap (C \cup B) \cap \bar{C}$ і визначити ступінь приналежності одного елемента множини D , використовуючи максимінний спосіб.



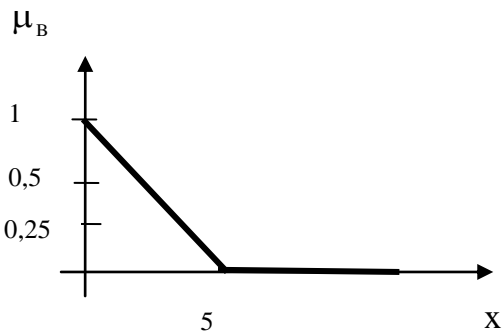
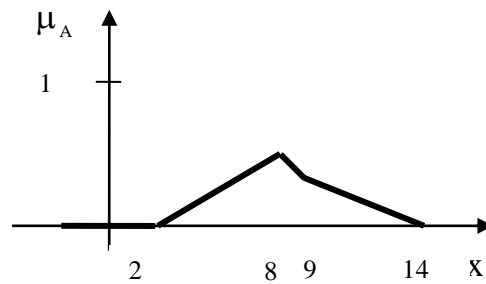
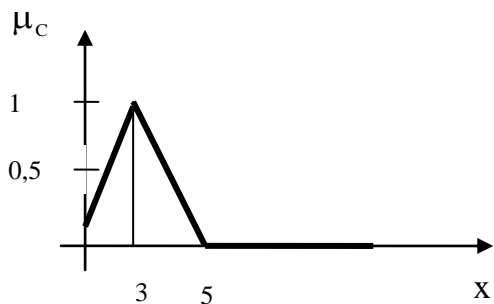
17) Дано 3 нечітких множини A , B , C (задані їх функції приналежності). Побудувати функцію приналежності нечіткої множини $D = \bar{A} \cap (C \cup B) \cap \bar{C}$ і визначити ступінь приналежності одного елемента множини D , використовуючи алгебраїчний спосіб.



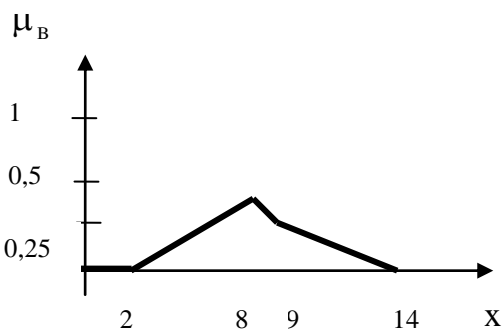
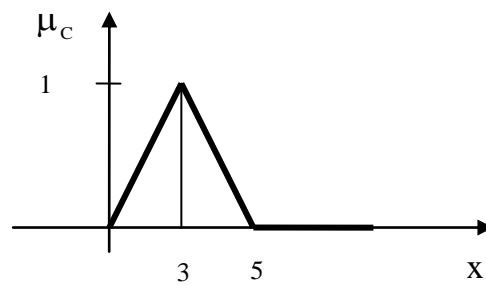
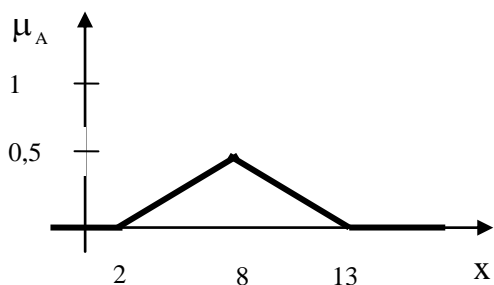
18) Дано 3 нечітких множини A , B , C (задані їх функції приналежності). Побудувати функцію приналежності нечіткої множини $D = \bar{A} \cap (C \cup B) \cap \bar{C}$ і визначити ступінь приналежності одного елемента множини D , використовуючи спосіб обмежень.



19) Дано 3 нечітких множини A, B, C (задані їх функції приналежності). Побудувати функцію приналежності нечіткої множини $D = \bar{A} \cup \bar{B} \cap \bar{C}$ і визначити ступінь приналежності одного елемента множини D , використовуючи алгебраїчний спосіб.



20) Дано 3 нечітких множини A, B, C (задані їх функції приналежності). Побудувати функцію приналежності нечіткої множини $D = \bar{A} \cup \bar{B} \cap \bar{C}$ і визначити ступінь приналежності одного елемента множини D , використовуючи спосіб обмежень.



3.3. Алгебра нечітких чисел. Нечіткі величини і нечіткі числа

Нечіткою величиною називається нечітка підмножина множини дійсних чисел. Найчастіше в практичних додатках використовуються не взаємодіючі нечіткі величини, коли прийняття обмежень для однієї з них не впливає на обмеження іншої, що аналогічно поняттю незалежних випадкових величин у теорії імовірності.

Нечітке число є моделлю невизначеної величини, значення якої "приблизно дорівнює m " чи "близько m ". Іноді у визначенні поняття нечіткого числа ознака нормальності (верхня границя функції приналежності досягає значення, що дорівнює 1, вважається несуттєвим.

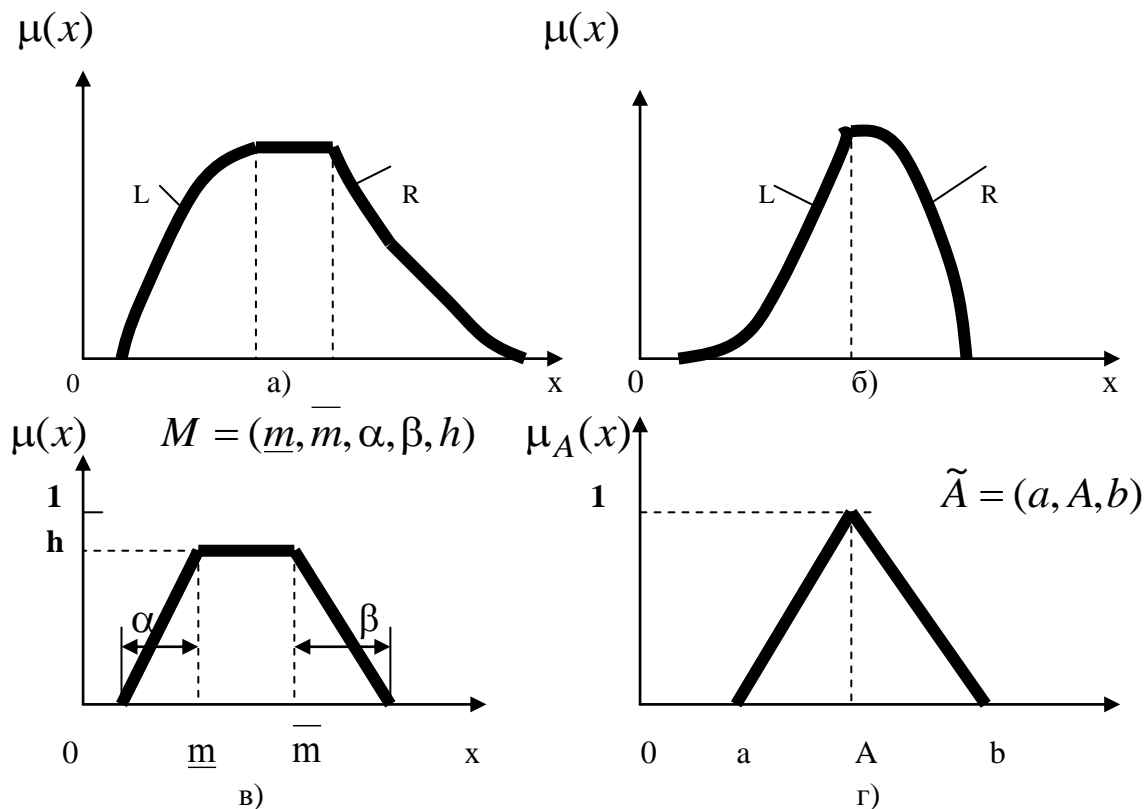


Рис. 3.17. Способи представлення нечітких чисел

Найбільш загальним є представлення нечіткого числа у вигляді нечіткого інтервалу з функцією приналежності, що має плоску вершину, монотонно зростаючу ліву гілку (L) і монотонно спадну праву гілку (R) (рис.3.17,а). Часто використовується трикутна форма представлення нечіткого числа (рис.3.17б).

Можливі й інші форми і способи представлення нечітких чисел, а також символічної форми їхнього запису. Так, наприклад, нечітке число \tilde{A} за рис. 3.17, г, можна записати у вигляді:

$$\tilde{A} = \int \mu_A(x) | x \text{ або } \tilde{A} = \int_a^A \frac{(x-a)}{(A-a)} | x + \int_A^b \frac{(b-x)}{(b-A)} | x,$$

де \int – символ об'єднання по всім x .

Відповідно до цієї символіки нечітке число $\tilde{2}$ (приблизно 2) можна представити у вигляді (рис. 3.18):

$$\tilde{2} = \int_1^2 (x-1) | x + \int_2^3 (3-x) | x, \text{ також } \tilde{2} = (a, A, b) = (1, 2, 3).$$

Алгебра нечітких чисел істотно спрощується, якщо вони задані лінійною регресійною формою (рис. 3.17, в,г). Трапеціє-подібна форма нечіткого числа може бути задана п'ятіркою дійсних чисел $(\underline{m}, \overline{m}, \alpha, \beta, h)$ (рис.3.17, в,г), трикутна форма (звичайно нормального числа) – трійкою (a, A, b) (рис.3.17, г).

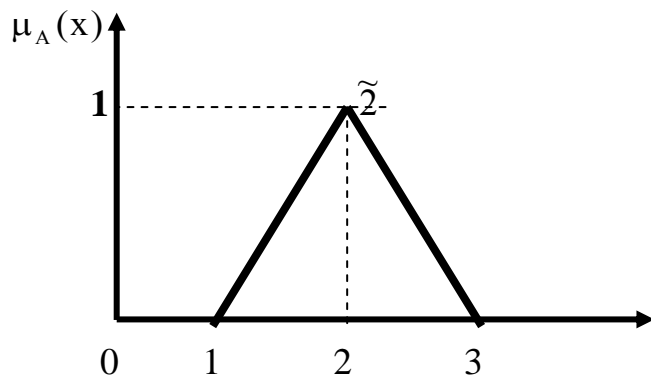


Рис. 3.18. Нечітке число $\tilde{2}$

3.3.1. Теоретичні основи операцій

При виконанні арифметичних дій над нечіткими числами варто мати на увазі три обставини:

1) чи є нечіткі числа не взаємодіючими, чи вони взаємодіють;

2) позитивним чи негативної є нечітке число;

3) чи є арифметична операція зростаючою (збільшувальною): додавання, множення, чи вона є спадною (зменшувальною), % вирахування, розподіл.

З погляду практичних додатків більший інтерес представляють зростаючі операції (додавання і множення) над не взаємодіючими позитивними (визначеними на R^+ – множина ненегативних дійсних чисел) нечіткими числами. Теоретичною основою арифметичних операцій на нечітких числах є принцип узагальнення Заді.

Основний зміст принципу узагальнення Заді в алгебрі нечітких чисел зводиться до такого. Нехай $y = f(x_1, x_2)$, де $y \in Y, x_1 \in X_1, x_2 \in X_2$. Задамо, використовуючи функції приналежності $\mu_{A_1}(x_1)$ і $\mu_{A_2}(x_2)$, нечіткі множини (нечіткі числа) A_1 і A_2 відповідно в X_1 і X_2 (у нас $X_1 = X_2 = R$). Тоді нечітка множина B (результат будь-якої арифметичної операції) у Y (у нас $Y = R$) тобто образ A_1 і A_2 визначитися відповідно до цього принципу функцією приналежності

$$\mu_B(y) = \max_{y=f(x_1, x_2)} \{ \min[\mu_{A_1}(x_1), \mu_{A_2}(x_2)] \}. \quad (3.1)$$

Якщо позначити символом \otimes будь-яку розширену бінарну арифметичну операцію над нечіткими числами M і N з носіями S_M і S_N , то функція приналежності результату C цієї операції визначитися як

$$\mu_C(y) = \mu_{M \otimes N}(y) = \max_{\substack{y=x_1 \otimes x_2 \\ x_1 \in S_M, x_2 \in S_N}} \{ \min[\mu_M(x_1), \mu_N(x_2)] \}. \quad (3.2)$$

Носій S_C нечіткого числа C можна знайти за правилами інтервальної арифметики, тому що

$$S_C = \{ y : y = x_1 \otimes x_2, x_1 \in S_M, x_2 \in S_N \}.$$

Якщо висоти функцій приналежностей нечітких чисел M і N не рівні, то при здійсненні арифметичних операцій над нечіткими числами слід ураховувати ефект, що зрізує (рис.3.19).

Необхідність вирівнювання висот функцій розподілу за мінімальним рівнем впливає як з аналітичних представлень (3.2) так і з графічного розгляду операцій (порівняй рівні α_1 і α_2 на (рис.3.19))

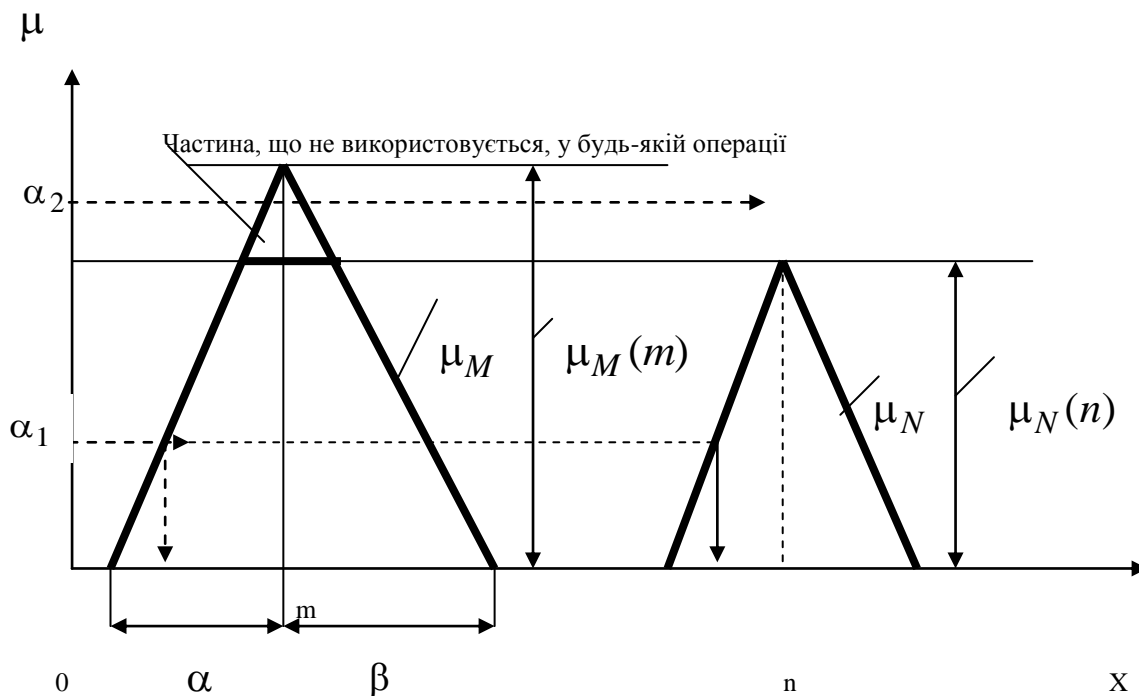


Рис. 3.19. Ефект, що зрізує

Максимальне значення $\mu_M(m)$ функції приналежності $\mu_M(x)$ нечіткого числа M (рис. 3.19) можна інтерпретувати як

ступінь надійності (ступінь кількості інформації, властива цьому максимуму), а розмахи α і β – як неточність визначення числа.

Залежно від форми представлення (опису) нечіткого числа можна створити велику кількість аналітичних і графічних алгоритмів операцій над нечіткими числами, однак, усі вони у своїй основі будуть містити один принцип – принцип узагальнення Заді. Ряд таких алгоритмів має свідомо наближений характер. Наприклад, представляючи числа трикутною формою (рис.3.17, г), результат їх добутку можна одержати в такій формі тільки приблизно.

Розглянемо представлення нечітких чисел у вигляді рис. 3.17, г. Відповідно до принципу узагальнення для двох нечітких чисел і результату будь-якої арифметичної операції (рис.3.20):

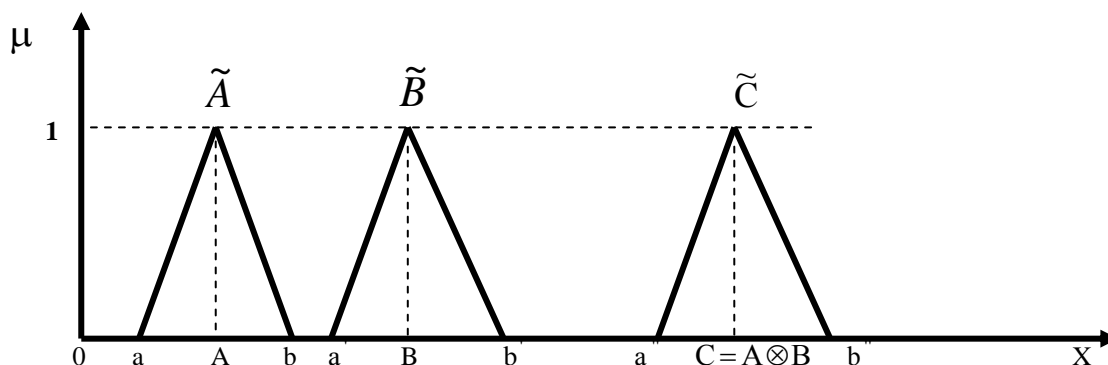


Рис.3.20. Результат будь-якої арифметичної операції

$$\begin{aligned} \tilde{C} = \tilde{A} \otimes \tilde{B} &= \int \min(\mu_A(x), \mu_B(y)) | x \otimes y = \left(\int_a^A \mu_A(x) | x + \int_A^b \mu_A(x) | x \right) \otimes \\ &\otimes \left(\int_{a'}^B \mu_B(x) | x + \int_B^{b'} \mu_B(x) | x \right) = \left(\int_{a''}^{A \otimes B} \mu_{A \otimes B}(x) | x + \int_{A \otimes B}^{b''} \mu_{A \otimes B}(x) | x \right), \end{aligned}$$

де \otimes – узагальнена розширена бінарна арифметична операція; a'', b'' одержують з a, b, a', b' залежно від конкретної операції, а функція $\mu_{A \otimes B}$ визначається конкретною операцією, так і умовою

нормування (нормування – вибір з рівноправних об'єктів якого-небудь одного, котрий можна дорівняти до 1 для спрощення рівнянь).

Розглянемо операцію додавання двох нечітких чисел.

$\tilde{A} \oplus \tilde{B} = \tilde{C}$; $a'' = a + a'$; $b'' = b + b'$; $C = A + B$ (за правилами інтервальної арифметики).

Функцію μ_C варто шукати у вигляді: $\mu_C = k_1 \cdot x + k_2$.

Виходячи з умови нормування, маємо

для $a'' \leq x \leq C$:

$$\begin{cases} k_1 C + k_2 = 1 \\ k_1 a'' + k_2 = 0 \end{cases} \Rightarrow k_1 = \frac{1}{C - a''}; k_2 = \frac{-a''}{C - a''}; \mu_C = \frac{x - a''}{C - a''}$$

для $C \leq x \leq b''$:

$$\begin{cases} k_1 C + k_2 = 1 \\ k_1 b'' + k_2 = 0 \end{cases} \Rightarrow k_1 = \frac{1}{C - b''}; k_2 = \frac{-b''}{C - b''}; \mu_C = \frac{b'' - x}{b'' - C}$$

У такий спосіб,

$$\tilde{A} \oplus \tilde{B} = \int_a^C \frac{x - a''}{C - a''} | x + \int_C^{b''} \frac{b'' - x}{b'' - C} | x = \tilde{C}.$$

Приклад 3.7. (рис.3.21)

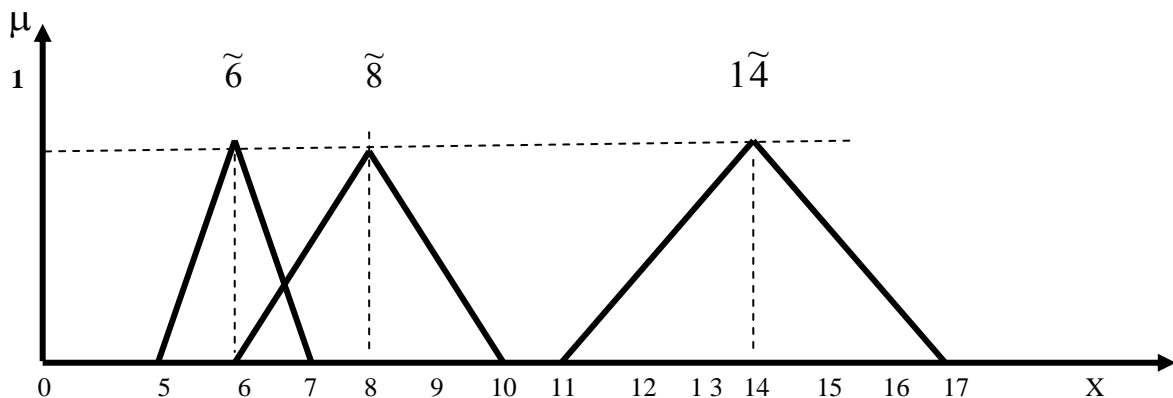


Рис.3.21. Додавання двох нечітких чисел

$$\begin{aligned} \tilde{6} \oplus \tilde{8} &= \left(\int_5^6 (x-5) | x + \int_6^7 (7-x) | x \right) \oplus \left(\int_6^8 \frac{(x-6)}{2} | x + \int_8^{10} \frac{(10-x)}{2} | x \right) = \\ & \int_{11}^{14} \frac{x-11}{14-11} | x + \int_{14}^{17} \frac{17-x}{17-14} | x = \int_{11}^{14} \frac{x-11}{3} | x + \int_{14}^{17} \frac{17-x}{3} | x = 1\tilde{4} \end{aligned}$$

Розглянемо операцію множення двох нечітких чисел \tilde{A} і \tilde{B} . Функцію μ_C результату $\tilde{C} = \tilde{A} \circ \tilde{B}$ шукаємо у вигляді $\mu_C = k_1 \cdot \sqrt{x} + k_2$. У результаті нескладних аналогічних перетворень з обліком нормування можна одержати:

$$\tilde{A} * \tilde{B} = \int_a^c \frac{\sqrt{x} - \sqrt{a''}}{\sqrt{c} - \sqrt{a''}} | x + \int_c^{b''} \frac{\sqrt{b''} - \sqrt{x}}{\sqrt{b''} - \sqrt{c}} | x = \tilde{C}.$$

Приклад 3.8. (рис.3.22)

$$\begin{aligned} \tilde{2} * \tilde{3} &= \left(\int_1^2 (x-1) | x + \int_2^3 (3-x) | x \right) \cdot \left(\int_2^3 (x-2) | x - \int_3^4 (4-x) | x \right) = \\ &= \int_2^6 \frac{\sqrt{x} - \sqrt{2}}{\sqrt{6} - \sqrt{2}} | x + \int_6^{12} \frac{\sqrt{12} - \sqrt{x}}{\sqrt{12} - \sqrt{6}} | x. \end{aligned}$$

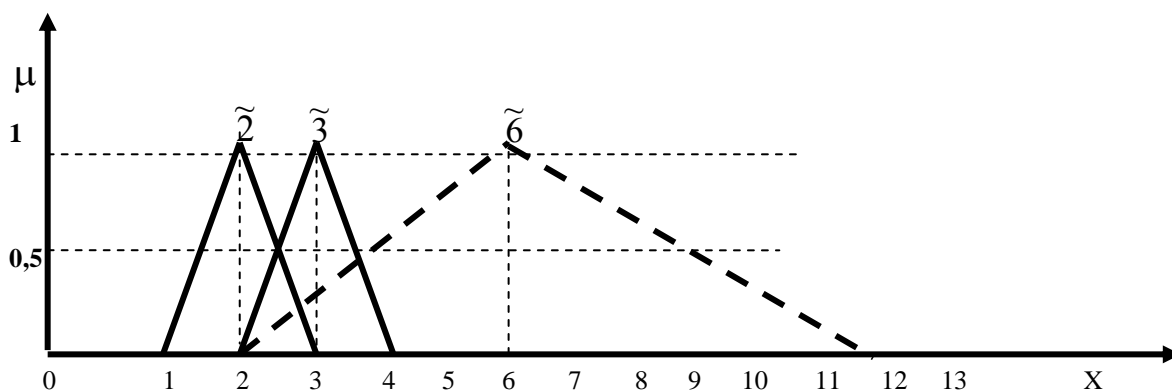


Рис. 3.22. Множення двох нечітких чисел

Обробка трапецієподібних нечітких інтервалів M_i і M_j за рис.3.23 дозволяє одержати в підсумку арифметичної операції додавання \oplus також трапецієподібний нечіткий інтервал M по формулах:

$$M_i = (\underline{m}_i, \bar{m}_i, \alpha_i, \beta_i, h_i);$$

$$M_j = (\underline{m}_j, \bar{m}_j, \alpha_j, \beta_j, h_j);$$

$$M = M_i \oplus M_j; M = (\underline{m}, \bar{m}, \alpha, \beta, h);$$

де

$$h = \min(h_i, h_j); \text{— ефект зрізу} \quad \alpha = h \cdot \left(\frac{\alpha_i}{h_i} + \frac{\alpha_j}{h_j} \right); \quad \beta = h \cdot \left(\frac{\beta_i}{h_i} + \frac{\beta_j}{h_j} \right);$$

$$\underline{m} = \underline{m}_i + \underline{m}_j - \alpha_i - \alpha_j + \alpha;$$

$$\bar{m} = \bar{m}_i + \bar{m}_j + \beta_i + \beta_j - \beta.$$

Приведені залежності дозволяють розглянути приклад, зв'язаний з оцінюванням джерел фінансування.

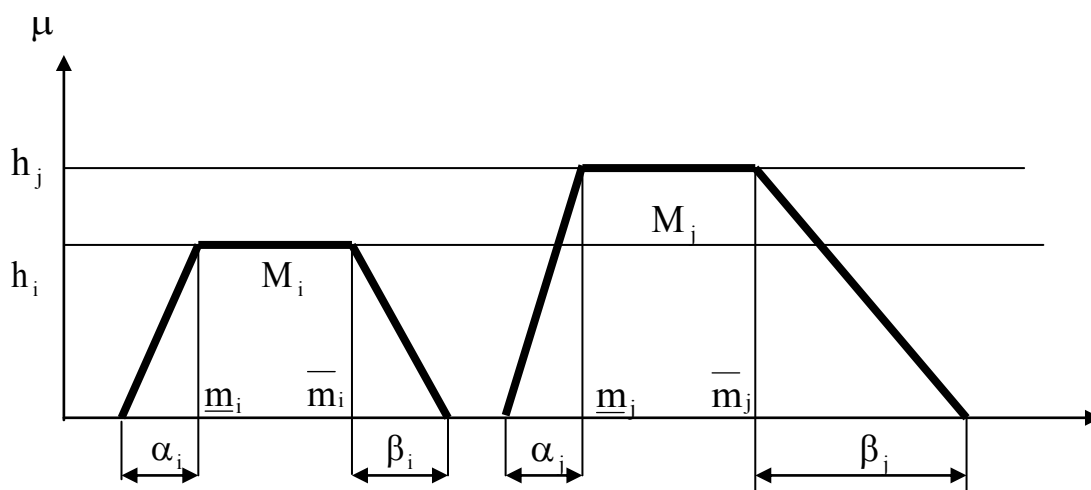


Рис.3.23. Обробка трапецієподібних нечітких інтервалів

Приклад 3.9. Оцінювання грошових коштів у бюджеті. У рамках складання проекту бюджету розглядаються різні джерела фінансування, причому деякі з них характеризуються неточністю

оцінки грошових сум на день оцінювання, а інші – малою надійністю. У прикладі беруться чотири джерела фінансування, що позначаються буквами А, У, С і D.

Джерело А: надійне і точне, очікувана сума 100 тис. доларів.

Джерело В: фінансування надійно забезпечується, його сума може змінюватися від 40 до 100 тис. залежно від кон'юнктури, але з найбільшою імовірністю можна чекати надходження розміром від 50 до 70 тис. доларів.

Джерело С: розумно думати, що фінансування буде надано і складе суму 100–110 тис. доларів, але рішення поки не прийнято і не можна цілком виключити варіант відмовлення від фінансування.

Джерело D: дуже ненадійне тому, що нове і хитливе. Можна чекати надходження розміром 20 тис. доларів і вище, але у будь-якому випадку не більше 30 тис. доларів.

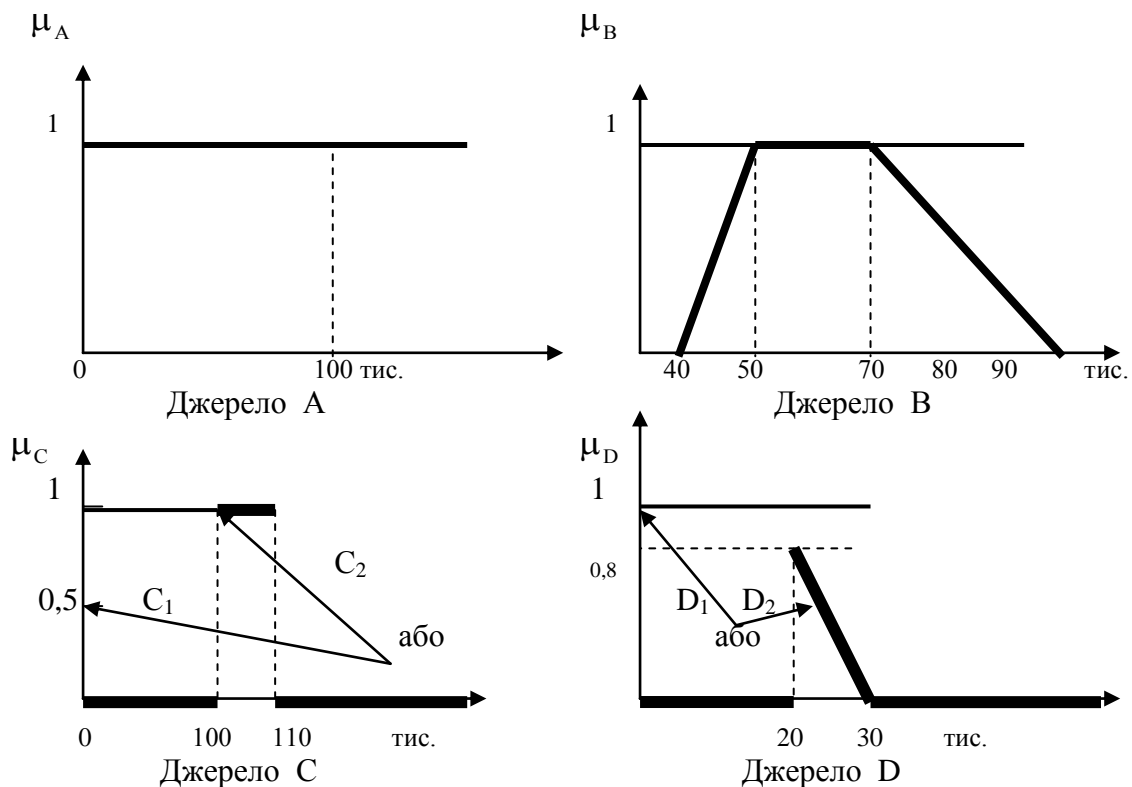


Рис.3.24. Різні джерела фінансування

Різні джерела фінансування можна представити за допомогою нечітких чисел з розподілами, зображеними на рис.3.24.

Кожна нечітка величина розглядається тут як об'єднання трапецієподібних і не обов'язково нормальних нечітких інтервалів. Кожний з цих нечітких інтервалів M_i представлений п'ятіркою

$$M_i = (\underline{m}_i, \bar{m}_i, \alpha_i, \beta_i, h_i),$$

де \underline{m}_i і \bar{m}_i – відповідно нижнє і верхнє модальні значення нечіткого інтервалу M_i ; α_i і β_i – лівий і правий коефіцієнти нечіткості, а висота нечіткого інтервалу. Відповідно до цих позначень нечіткі величини, пов'язані з різними джерелами фінансування, представляються у вигляді:

$$A = (100, 100, 0, 0, 1); \quad B = (50, 70, 10, 30, 1);$$

$$C = C_1 \cup C_2 = (0, 0, 0, 0, 0.5) \cup (100, 110, 0, 0, 1) –$$

дві можливості, об'єднані союзом АБО;

$$D = D_1 \cup D_2 = (0, 0, 0, 0, 1) \cup (20, 20, 0, 10, 0.8).$$

Сума в бюджеті виходить як об'єднання чотирьох варіантів:

$$S = (A \oplus B \oplus C_1 \oplus D) \cup (A \oplus B \oplus C_1 \oplus D_2) \cup$$

$$\cup (A \oplus B \oplus C_2 \oplus D_1) \cup (A \oplus B \oplus C_2 \oplus D_2) = S_1 \cup S_2 \cup S_3 \cup S_4.$$

Визначимо суму $A \oplus B$ за вищенаведеними формулами:

$$h = \min(h_1, h_2) = \min(1, 1) = 1;$$

$$\alpha = h \cdot \left(\frac{\alpha_1}{h_1} + \frac{\alpha_2}{h_2} \right) = 1 \cdot \left(\frac{0}{1} + \frac{10}{1} \right) = 10; \quad \beta = h \cdot \left(\frac{\beta_1}{h_1} + \frac{\beta_2}{h_2} \right) = 1 \cdot \left(\frac{0}{1} + \frac{30}{1} \right) = 30;$$

$$\underline{m} = \underline{m}_1 + \underline{m}_2 - \alpha_1 - \alpha_2 + \alpha = 100 + 50 - 0 - 10 + 10 = 150;$$

$$\bar{m} = \bar{m}_1 + \bar{m}_2 + \beta_1 + \beta_2 - \beta = 100 + 70 + 0 + 30 - 30 = 170.$$

Тоді $A \oplus B = (150; 170; 10; 30; 1)$.

Роблячи аналогічно далі, одержимо:

$$(A \oplus B) \oplus C_1 = (145; 185; 5; 15; 0; 0.5);$$

$$S_1 = [(A \oplus B) \oplus C_1] \oplus D_1 = (145; 185; 5; 15; 0.5);$$

$$S_2 = [(A \oplus B) \oplus C_1] \oplus D_2 = (165; 209; 5; 21; 0.5);$$

$$S_3 = [(A \oplus B) \oplus C_2] \oplus D_1 = (250; 280; 10; 30; 1);$$

$$S_4 = [(A \oplus B) \oplus C_2] \oplus D_2 = (268; 306; 8; 34; 0.8).$$

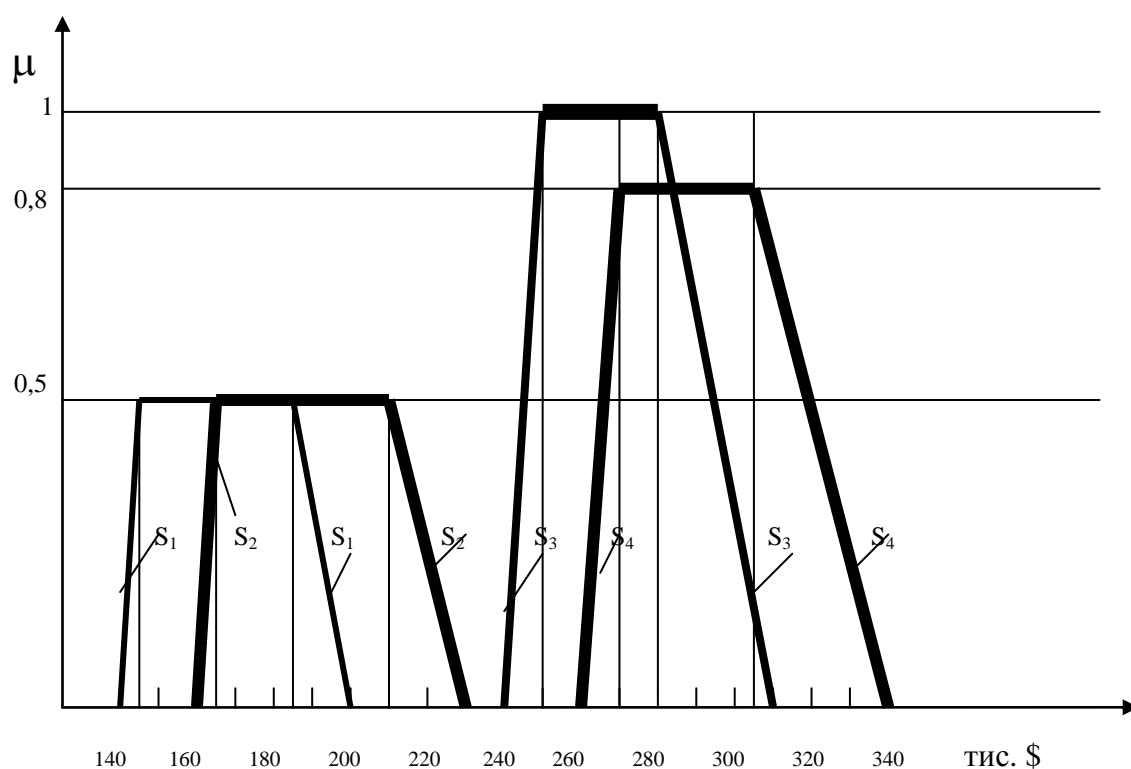


Рис. 3.25. Чотири варіанти фінансування S_1, S_2, S_3, S_4

На рис. 3.25 показані чотири варіанти фінансування S_1, S_2, S_3, S_4 і їхнє об'єднання. Відповідно до отриманого результату область найбільш ймовірного фінансування простирається в діапазоні 250–280 тис. доларів; перевищення суми в 280 тис. доларів можливе, але менш ймовірне (рівень 0,8); малоймовірно і те, що надходження не складе більш 150–200 тис. доларів (рівень 0,5). У будь-якому випадку вони не можуть опуститися нижче від 140 тис. чи піднятися вище від 340 тис. доларів.

Відзначимо, що рівні 0,5 і 0,8 при описі джерел С і D, виглядають трохи довільними, але вказують, що більш ймовірно (зі ступенем 0,8) одержати 20 тис. доларів від джерела D, ніж нічого не одержати від джерела З (зі ступенем 0,5). Ці значення не згортаються в одне і знову з'являються в кінцевому результаті. Отже, зовсім не обов'язково знати їх точно. Важливо саме те, що вони різні і можуть служити свого роду мітками, що полегшують тлумачення кінцевого результату.

3.4. Лінгвістичні змінні. Нечітка і лінгвістична змінна

Структура нечіткої змінної може бути представлена у вигляді: $\langle \alpha, X, C \rangle$, де α – найменування нечіткої змінної, її ім'я (символ); $X = \{x\}$ – область її визначення (універсальна множина, є областю міркувань); $C = \bigcup_{x \in X} \mu_{\alpha}(x)$ – сукупність нечітких множин

x , що описує обмеження на можливі значення нечіткої змінної В якості прикладу нечіткої змінної розглянемо нечітку множину, яка буде характеризувати "гарячу каву". У цьому випадку відповідна нечітка змінна може бути представлена таким чином:

$$\langle \text{Гаряча кава}, \{x | 0^{\circ}\text{C} < x < 100^{\circ}\text{C}\}, B \rangle$$

де $B = \{x, \mu_B(x)\}$ – нечітка множина з функцією належності $\mu_B(x)$, яка може бути задана, зокрема, графічно. Таким чином, основою поняття нечіткої змінної є нечітка множина. Додатково запроваджується зазначення імені (символу) цієї нечіткої множини й універсальної множини. У такому розумінні α – символ нечіткої множини С, визначеної на універсальній множині X.

Якщо значеннями нечіткої змінної стають речення або слова спеціальною (формальною) мовою, то такі нечіткі змінні Заде називає *лінгвістичними змінними*. Прикладом лінгвістичної

змінної є нечітка змінна ВИСОТА. Її значення можуть бути утворені поняттями: ВИСОКИЙ, НИЗЬКИЙ, СЕРЕДНІЙ та додатковими словами: НЕ ДУЖЕ, ДОСИТЬ, БІЛЬШЕ АБО МЕНШЕ і т.ін. Лінгвістичні змінні призначені для характеристики складних і погано визначених явищ і систем, для яких не підходять звичайні кількісні оцінки.

Структура лінгвістичної змінної у загальному вигляді може бути представлена:

$$\langle \beta, T, X, G, M \rangle \quad (3.1)$$

наприклад: $\langle \text{ВАРТИСТЬ}, T, [0, 5000], G, M \rangle$,

де β – найменування лінгвістичної змінної; у наведеному прикладі це ВАРТИСТЬ; $T = \{T_i\}$ – множина значень лінгвістичної змінної (базова терм-множина); X – універсальна множина (область міркувань), що є областю визначення значень лінгвістичної змінної; G – граматики (синтаксична процедура, яка описує процес утворення або генерування з множини T нових, осмислених у розглянутому контексті значень для даної лінгвістичної змінної; M – семантична процедура, яка дозволяє поставити у відповідність кожному новому значенню даної лінгвістичної змінної, отриманої за допомогою процедури G , деякий осмислений зміст за допомогою формування відповідного нечіткої множини.

Базова терм-множина T

Базова терм-множина $T = \{T_i\}$ являє собою найменування значень лінгвістичної змінної на природній або формальній мові, областю визначення яких є множина X . Потужність множини T , виходячи з практичних міркувань, не повинна перевищувати 5...9.

Кожен терм – це слово або речення. Для наведеного вище прикладу трьохелементна терм-множина може бути побудована, наприклад, таким чином:

$$T = \left\{ \begin{array}{l} T_1, T_2, T_3 \\ \text{мала середня висока} \end{array} \right\},$$

Кожному терму T_i , тобто кожному значенню лінгвістичної змінної, відповідає нечітка змінна $\langle T_i, X, C_i \rangle$, де C_i – нечітка множина з функцією приналежності $\mu_{T_i}(x)$ (рис.3.26).

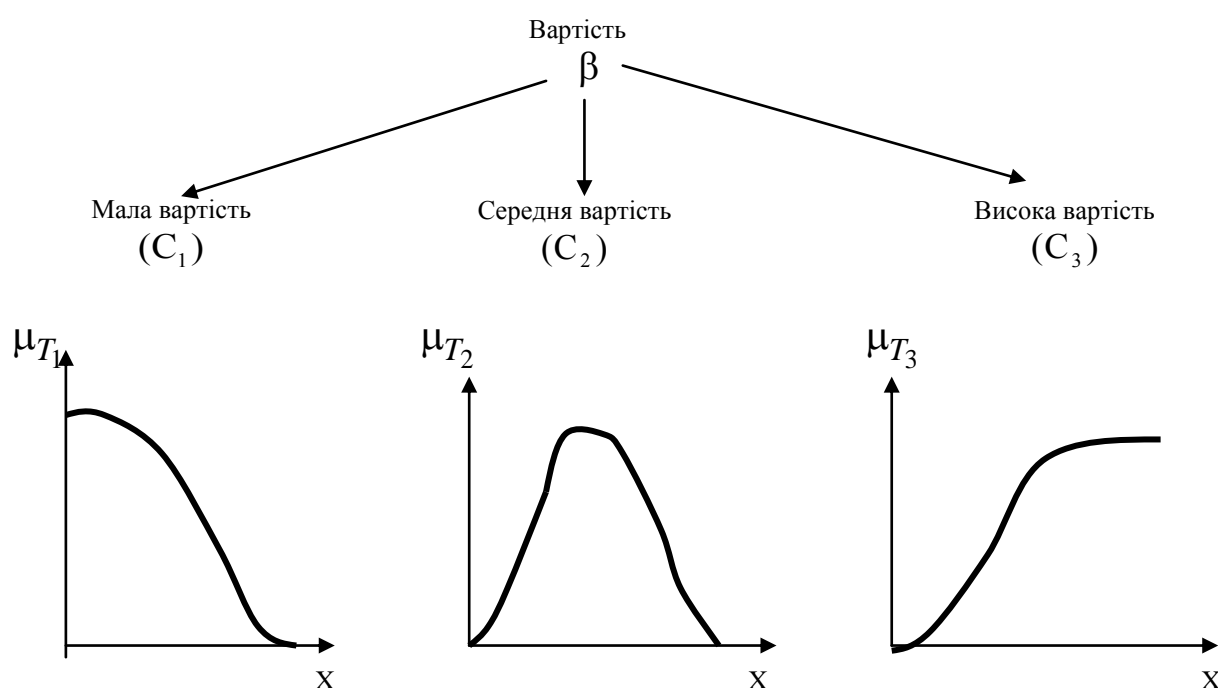


Рис.3.26. Терм-множини лінгвістичної змінної вартості

Таким чином, значення T_i лінгвістичної змінної з найменуванням β , у свою чергу, символами (іменами) нечітких підмножин C_i універсальної множини X , повністю описуваних своїми функціями приналежності μ_{T_i} .

Часто терм (символ) T_i і відповідну йому нечітку множину C_i ототожнюють, якщо це не викликає плутанину. Наприклад, якщо

$$\mu_{\text{МОЛОДИЙ}}(x) = \begin{cases} 1, & \text{якщо } x \leq 25 \\ \left(1 + \left(\frac{x-25}{5}\right)^2\right)^{-1}, & \text{якщо } x > 25, \end{cases}$$

то нечітку множину $C_i \subseteq X$, позначену як МОЛОДИЙ, можна представити у вигляді

$$T_i = C_i = \text{МОЛОДИЙ} = \int_0^{25} 1 |x + \int_{25}^{100} \left(1 + \left(\frac{x-25}{5}\right)^2\right)^{-1} |x.$$

Універсальна множина лінгвістичної змінної X є областю визначення значень лінгвістичної змінної. Якщо максимальна вартість не перевищує, скажімо, 5000 тис. грн., множину X визначено континуумом $[0, 5000]$. Можливий вид функцій приналежності на цій універсальній множині показаний на рис.3.27.

Залежно від характеру множини X лінгвістичні змінні можуть бути розділені на числові і нечислові.

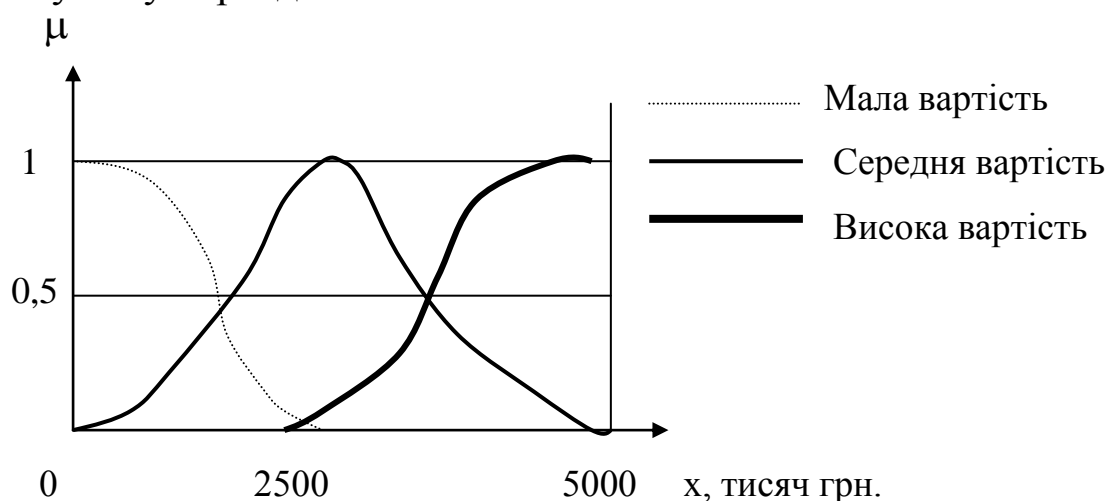


Рис. 3.27. Можливий вид функцій приналежності вартості

Число називається лінгвістичною змінною, для якої область визначення X є підмножиною числовій осі. Нечіткі змінні, відповідні значенням числової лінгвістичної змінної представ-

ляються нечіткими числами. Прикладом нечислової лінгвістичної змінної є змінна КОМПЕТЕНТНІСТЬ зі значеннями: КОМПЕТЕНТНИЙ, ВИСОКО КОМПЕТЕНТНИЙ, НЕДОСТАТНЬО КОМПЕТЕНТНИЙ і т.ін. Але в цьому випадку можна побудувати нечіткі множини на універсальній множині U , позначеній деякими символами вражень, т. зв. "психологічному континуумі" (рис.3.28).

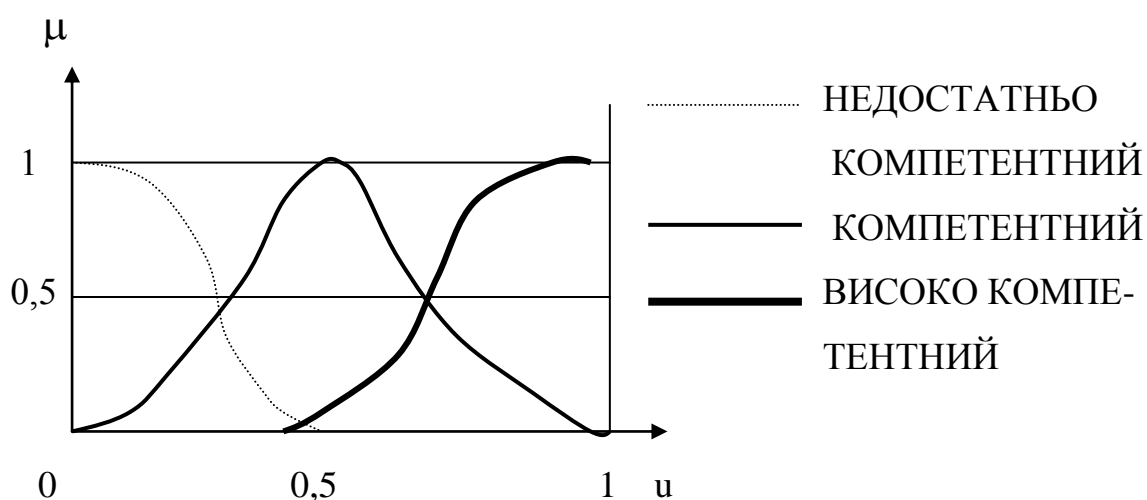


Рис. 3.28. Числова лінгвістична змінна

Синтаксична процедура G

Синтаксична процедура G (зокрема, формальна граматика) описує процес утворення нових, осмислених для даної задачі значень лінгвістичної змінної, виходячи з її базової терм-множини T . За своєю суттю це процедура розширення базової терм-множини. Для (3.1) така синтаксична процедура повинна, наприклад, утворювати нові значення: ДУЖЕ МАЛА ВАРТІСТЬ, ДУЖЕ ВИСОКА ВАРТІСТЬ, НЕЙМОВІРНО ВИСОКА ВАРТІСТЬ і т.ін. Такі складові терміни повинні бути генеровані позбавленою контексту граматикою.

Іноді процедура G полягає в простому перерахуванні можливих термів (як і звичайне множину можна задати не тільки породжує процедурою, але і списком).

Грамматична структура складових термінів значень лінгвістичної змінної має 4 категорії елементів:

1) первинні терміни T_i базової терм-множини $T = \{T_i\}$, наприклад, ВИСОКИЙ;

2) заперечення НЕ, сполучники І, АБО;

3) модифікатори або лінгвістичні невизначеності типу ДУЖЕ, БАГАТО, МАЛО, БІЛЬШЕ, МЕНШЕ, ЦІЛКОМ, НАДЗВИЧАЙНО тощо, що позначаються символом h ;

4) маркери-роздільники (прогалини), увідні слова. Синтаксичну процедуру G (формальну граматику) можна розглядати як оператор, діючий на базову терм-множину, що призводить до утворення розширеної терм-множини T^* лінгвістичної змінної $T^* = T \cup G(T)$.

Семантична процедура M

Семантична процедура M ставить у відповідність терміну T_i нечітку множину C_i , тобто $M(T_i) = C_i$.

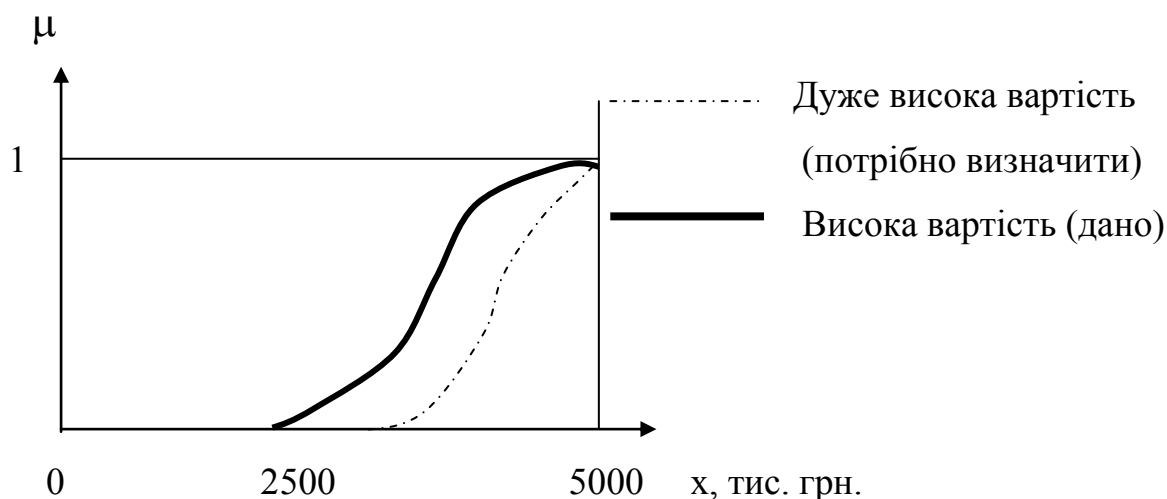


Рис.3.29. Дуже висока вартість

Семантична процедура M дозволяє також перетворити кожне нове значення лінгвістичної змінної, утворене синтаксичною процедурою G (наприклад, ДУЖЕ ВИСОКА

ВАРТІСТЬ), нечітке формування нечіткої множини (рис.3.29). Ця процедура часто пов'язана з експертним опитуванням.

При визначенні семантики нових значень розширеної термножини T^* , лінгвістична змінна використовує раніше описані операції (об'єднання, перетинання та доповнення) над нечіткими множинами, а також деякі нові операції, використовувані тільки для лінгвістичних змінних (концентрації, розтягування і т.ін.) при обліку лінгвістичних невизначеностей.

Так, за змістом операцій \cup , \cap і доповнення маємо:

$$\begin{aligned} M(T_i \text{ і } T_j) &= C_i \cap C_j; \\ M(T_i \text{ або } T_j) &= C_i \cup C_j; \\ M(\text{не } T_i \text{ не}) &= \bar{C}_i; M(T_i) = C_i. \end{aligned} \quad (3.2)$$

Зауважимо тут, що якщо терм T_i – це слово або речення, то $M(T_i)$ – це значення, сенс терма у вигляді нечіткої множини C_i .

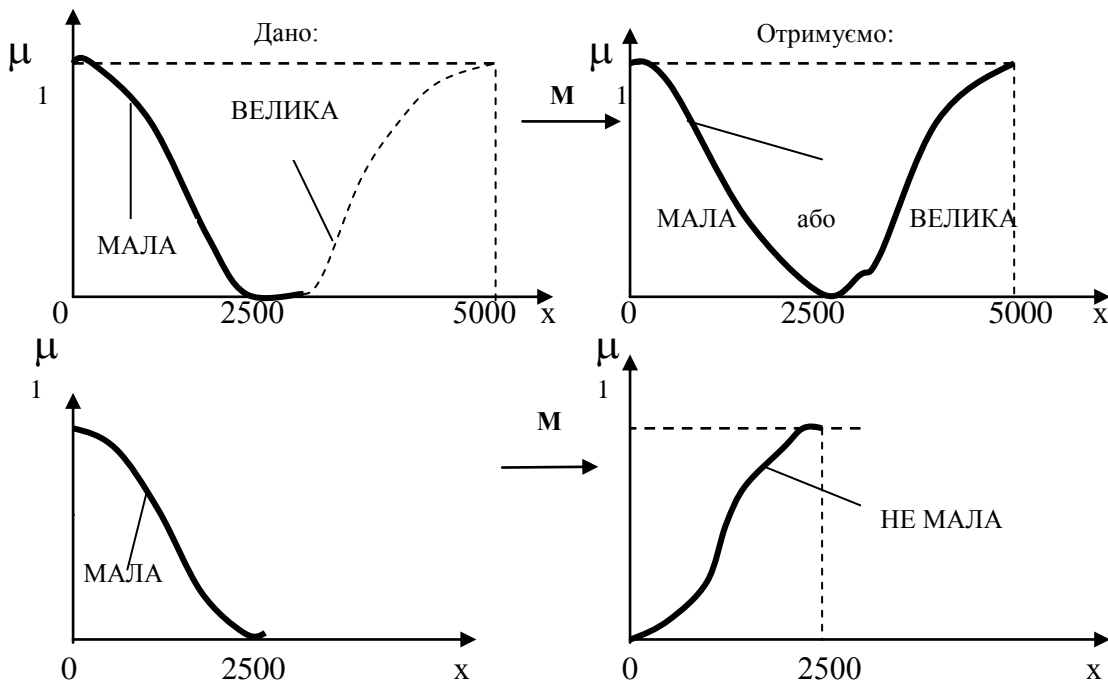


Рис.3.30. Операції над множинами

Операції над множинами C_i і C_j зводиться до перетворення функцій належності (рис.3.30).

Облік лінгвістичних невизначеностей необхідний у разі обчислення значення складеного терміну. Структуру складеного (розширеного) терміна можна визначити як

$$T_i^* = h \cdot T_i,$$

де h – модифікатор (лінгвістична невизначеність), T_i – термін базової терм-множини.

Наприклад: $h = \text{ДУЖЕ}$; $T_i = \text{ВИСОКИЙ}$; $T_i^* = h \cdot T_i = \text{ДУЖЕ ВИСОКИЙ}$.

Лінгвістичну невизначеність h можна розуміти як оператор, що переводить нечітку множину $C_i^* = M(hT_i)$, представляє значення $T_i^* = hT_i$.

Таким чином, лінгвістичні невизначеності h виконують функцію генерації великої множини значень лінгвістичної змінної з невеликого набору первинних термінів $T = \{T_i\}$.

Для визначення лінгвістичної невизначеності h як оператора, зручно використовувати спеціальні операції над нечіткими множинами (концентрації, розтягування і т. ін.). Так у повсякденному використанні лінгвістична невизначеність ДУЖЕ діє як підсилювач, генеруючи підмножину тієї множини, до якої вона застосовується.

Спеціальна операція, що має таку властивість, це операція концентрації:

$$CON(A) = A^2.$$

У результаті її застосування до множини A зменшується ступінь належності елементів даної множини, причому для елементів з високим ступенем приналежності це зменшення порівняно мале, а для елементів з малим ступенем приналежності відносно велике.

Стосовно до лінгвістичної невизначеності:

$$\text{ДУЖЕ} \quad T_i = T_i^2$$

або більш виразно

$$\text{ДУЖЕ} \quad T_i = \int_x \mu_{T_i^2}(x) | x.$$

Наприклад, якщо (рис.3.26)

$$C_i = (T_i) = \text{ВИСОКА} \quad \text{ВАРТІСТЬ} = \int_{2500}^{5000} \left(1 + \left(\frac{x-2500}{250} \right)^{-2} \right)^{-1} | x,$$

то

$$C_i = (T_i) = hT_i = \text{ДУЖЕ} \quad \text{ВИСОКА} \quad \text{ВАРТІСТЬ} = \int_{2500}^{5000} \left(1 + \left(\frac{x-2500}{250} \right)^{-2} \right)^{-2} | x.$$

При цьому ступінь приналежності елемента, скажімо, $x = 3000$ до класу ВИСОКА ВАРТІСТЬ дорівнює

$$\left(1 + \left(\frac{3000-2500}{250} \right)^{-2} \right)^{-1} = 0,8,$$

а ступінь приналежності того ж елемента до класу ДУЖЕ ВИСОКА ВАРТІСТЬ дорівнює $0,8^2 = 0,64$ (рис.3.31). У той же час ступені приналежності елемента $x = 4500$ до класів ВИСОКА ВАРТІСТЬ і ДУЖЕ ВИСОКА ВАРТІСТЬ відрізняються незначно (відповідно 0,98 та 0,96).

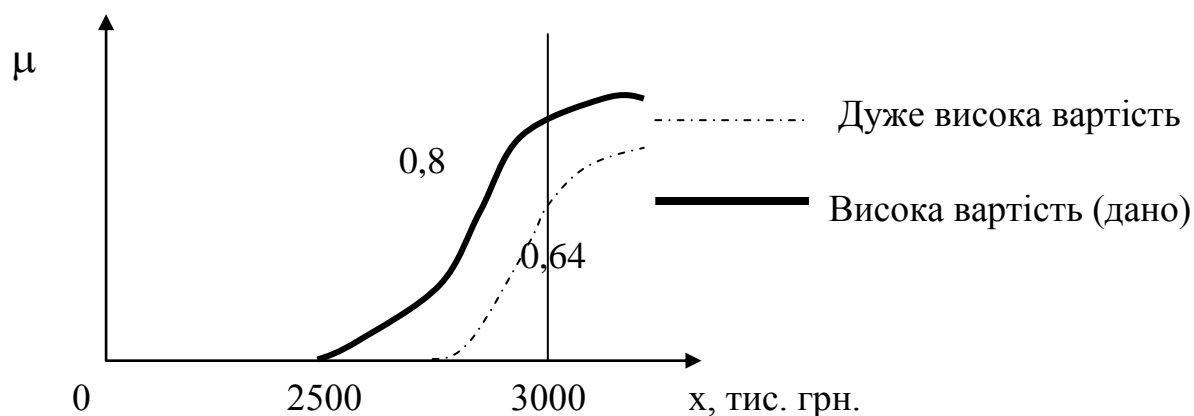


Рис.3.31. Дуже висока вартість

Розглянутий оператор може поєднуватися з самим собою. Так, ДУЖЕ ДУЖЕ $T_i = (\text{ДУЖЕ } T_i)^2 = T_i^4$.

У тому ж прикладі ступінь приналежності $x = 3000$ до класу ДУЖЕ ВИСОКА ВАРТІСТЬ складає $0,64^2 = 0,41$.

Таким чином, додатково до (3.2) семантика термів $T_i^* \in T^*$, що задається процедурою $M(T_i) = C_i$:

$$M(\text{ДУЖЕ } T_i) = C_i^2;$$

$$M(\text{ДУЖЕ ДУЖЕ } T_i) = C_i^4;$$

$$M(\text{НЕ ДУЖЕ } T_i) = (\bar{C}_i)^2.$$

При обчисленні значення складеного лінгвістичного терміна, що виключає крім лінгвістичної невизначеності h сполучники (зв'язки) І, АБО, НЕ, використовуються звичайні правила старшинства (черговості), що діють при перетворенні булевих формул:

Попередні (черговість)	Операція
1	h , НЕ
2	І
3	АБО

3.5. Нечіткі правила виводу

Нечіткий вивід займає центральне місце в нечіткій логіці і системах нечіткого управління. Процес нечіткого виводу являє собою деяку процедуру або алгоритм отримання нечітких виводів на основі нечітких умов і передумов. Цей процес поєднує в собі всі основні концепції теорії нечітких множин: функції приналежності, лінгвістичні змінні, нечіткі логічні операції, методи нечіткої імплікації і нечіткої композиції.

Нечітке лінгвістичне висловлювання. Нечітким лінгвістичним висловлюванням називають висловлювання таких видів:

1. Вислів « $\beta \in \alpha$ », де β – найменування лінгвістичної змінної, α – її значення, якому відповідає окремий лінгвістичний терм з базової терм-множини T лінгвістичної змінної β .

2. Вислів « $\beta \in \nabla \alpha$ », де ∇ – модифікатор, відповідний таким словам, як: «ДУЖЕ», «БІЛЬШЕ АБО МЕНШЕ», «БІЛЬШЕ» та інші.

3. Складові висловлювання, утворені з висловлювань видів 1 і 2 та нечітких логічних операцій у формі зв'язок: «І», «АБО», «ЯКЩО — ТО», «НЕ».

Приклад 3.10. Розглянемо деякі приклади нечітких висловлювань. Перше з них — *"швидкість автомобіля висока"* являє собою нечіткий вислів першого виду, в рамках якого лінгвістичній змінній *"швидкість автомобіля"* присвоюється значення *"висока"*. При цьому передбачається, що на універсальній множині X змінній *"швидкість автомобіля"* визначено відповідний лінгвістичний терм *"висока"*, який задається у формі функції належності деякого нечіткого. Нечіткий вислів другого виду *"швидкість автомобіля дуже висока"* означає, що лінгвістичній змінній *"швидкість автомобіля"* присвоюється значення *"висока"* з модифікатором *"ДУЖЕ"*, який змінює значення відповідного лінгвістичного терма *"висока"* на основі використання деякої розрахункової формули, наприклад, для операції концентрації $\text{CON}(\tilde{A})$ нечіткої множини для терма *"висока"*. Нечіткий вислів другого виду *"швидкість автомобіля більш або менш висока"* означає, що лінгвістичній змінній *"швидкість автомобіля"* присвоюється значення *"висока"* з модифікатором *"БІЛЬШЕ ЧИ МЕНШЕ"*, який змінює значення відповідного лінгвістичного терма *"висока"* на основі використання деякої розрахункової формули, наприклад

для операції розтягування $DIL(\tilde{A})$ нечіткої множини для терма "висока". Нижче, на рис. 3.32, зображений приклад функції належності терм-множини "середня" лінгвістичної змінної "швидкість автомобіля" (а) і визначення значень функцій належності цієї ж терм-множини для модифікаторів "ДУЖЕ" (б) і "БІЛЬШЕ-МЕНШЕ" (в).

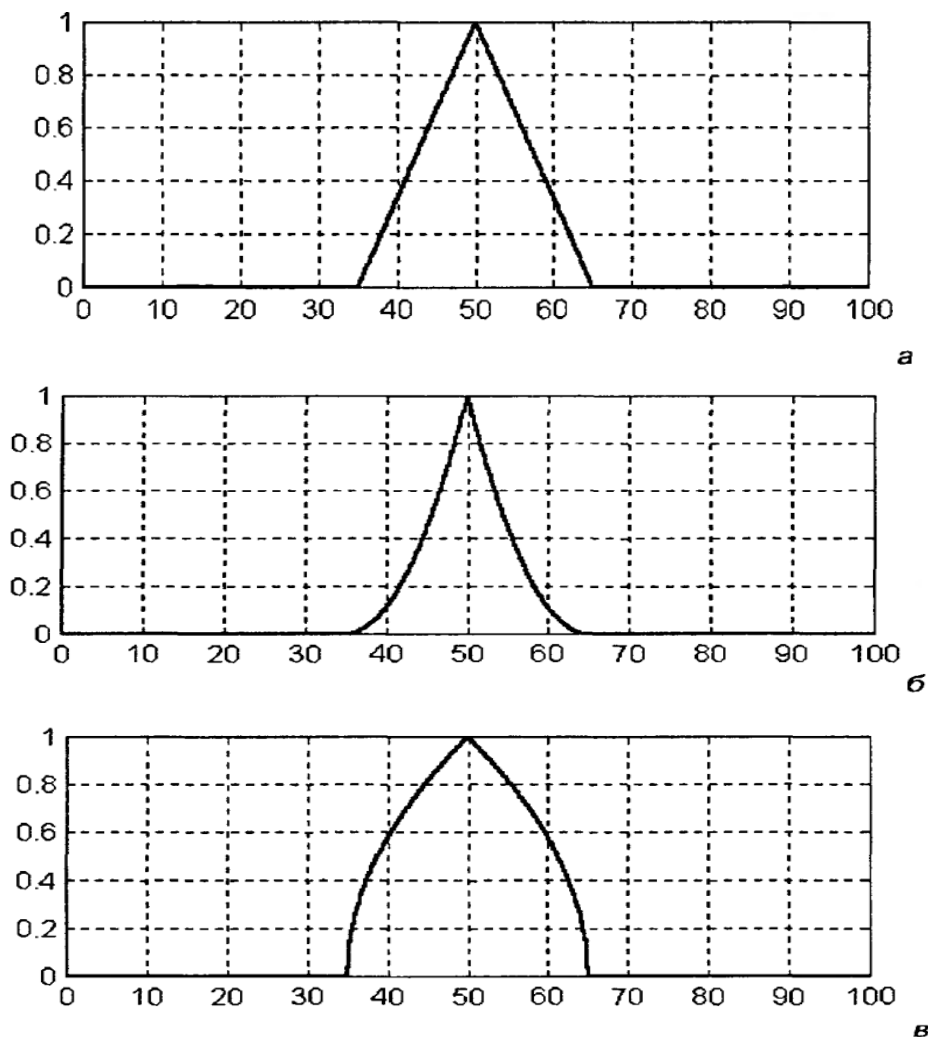


Рис. 3.32. Застосування модифікаторів "ДУЖЕ" (б) і "БІЛЬШ МЕНШ" (в) до терму "середня швидкість" (а)

Нарешті, нечітке висловлювання третього виду "швидкість автомобіля висока і відстань до перехрестя близька" означає, що одній лінгвістичній змінній "швидкість автомобіля"

присвоюється значення "висока", а інший лінгвістичній змінній "відстань до перехрестя" присвоюється значення "близька". Ці нечіткі висловлювання першого виду з'єднані логічною операцією нечітка кон'юнкція (операцією нечітке "І").

Базове правило виводу типу "якщо — то" (англ.: if – then rule) називається також *нечіткою імплікацією*, що набуває форми

якщо x це A , то y це B ,

де A і B – це лінгвістичні значення, ідентифіковані нечітким способом через відповідні функції приналежності для змінних x і y . Частина " x це A " називається умовою (передумовою), а " y це B " – наслідком (висновком).

Імплікацію якщо x це A , то y це B можна записати у скороченому вигляді $A \rightarrow B$.

Нечітке міркування – це процедура, яка дозволяє визначити вивід, що випливає з множини правил "якщо – то". Така множина при змінних може мати вигляд

якщо x_1 це A_1 і x_2 це A_2 і ... і x_n це A_n , то y це B .

Змінні x_1, x_2, \dots, x_N утворюють N -мірний вектор x , що становить аргумент умови, в якій A_1, A_2, \dots, A_N і B позначають величини відповідного коефіцієнта приналежності $\mu_A(x_i)$ і $\mu_B(y)$. Необхідно звернути увагу, що тут присутні індивідуальні функції належності для кожної змінної x_i і окремо для y . Випадкове значення функції приналежності $\mu_A(x)$, де x – це вектор $x = [x_1, x_2, \dots, x_N]$ відноситься до умови імплікації (рівень активації правила), має в подальшому інтерпретуватися з використанням введених раніше нечітких операцій. Можлива інтерпретація у формі логічного добутку множини або у формі алгебраїчного добутку:

– інтерпретація у формі логічного добутку

$$\mu_A(x) = \min \left(\mu_A(x_i) \right)_{i=1, \dots, N}$$

– інтерпретація у формі алгебраїчного добутку

$$\mu_A(x) = \prod_{i=1}^N \mu_A(x_i).$$

Приписування єдиного значення функції приналежності, що описує багатовимірну умову, будемо називати *агрегуванням передумови*. Кожній імплікації $A \rightarrow B$, що визначається виразом (A), можна приписати також єдине значення функції належності $\mu_{A \rightarrow B}(x, y)$. Найбільш популярні інтерпретації цієї функції також мають форму логічного чи алгебраїчного добутку:

– класична нечітка імплікація, запропонована Л. Заде

$$\mu_{A \rightarrow B} = \max \{ \min \{ \mu_A(x), \mu_B(y) \}, 1 - \mu_A(x) \},$$

– форма логічного добутку (нечітка імплікація, запропонована Мамдані)

$$\mu_{A \rightarrow B} = \min \{ \mu_A(x), \mu_B(y) \},$$

– форма алгебраїчного добутку

$$\mu_{A \rightarrow B} = \mu_A(x) \cdot \mu_B(y).$$

Приписування єдиного значення функції приналежності всієї імплікації будемо називати процедурою *агрегування* на рівні імплікації.

Системи нечіткого виводу призначені для перетворення значень вхідних змінних процесу управління на вихідні змінні на основі використання нечітких правил продукцій. Для цього системи нечіткого виводу повинні містити базу правил нечітких продукцій, реалізувати нечіткий висновок висновків на основі посилок або умов, представлених у формі нечітких лінгвістичних висловлювань.

У загальному випадку система нечіткого логічного висновку може бути представлена так, як це показано на рис.3.33.

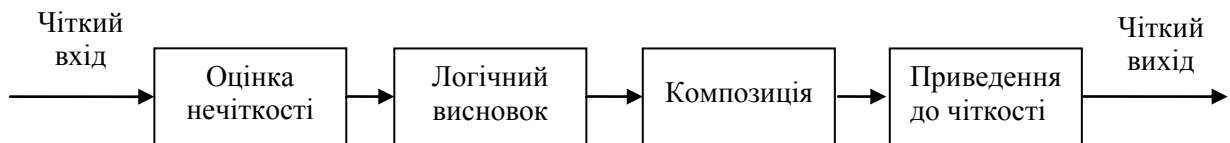


Рис.3.33. Схема нечіткого виводу

1. **Приведення до нечіткості** (фазифікація): функції належності, визначені на входних змінних, застосовуються до їх фактичних значень для визначення ступеня істинності кожної передумови кожного правила.

2. **Логічний висновок**: обчислене значення істинності для передумов кожного правила застосовується до висновків кожного правила. Це призводить до одної нечіткої множини, яка буде призначена кожній змінній виводу для кожного правила. Це призводить до одної нечіткої підмножини, яка буде призначена кожній змінній виводу для кожного правила. В якості правил логічного виводу зазвичай використовуються тільки операції *min* або *prod* (множення). В логічному висновку мінімуму функція належності «відсікається» по висоті, відповідній обчисленому ступеню істинності передумови правила (нечітка логіка «I»). В логічному висновку множення функція належності висновку масштабується за допомогою обчисленого ступеня істинності передумови правила.

3. **Композиція**: виходи всіх правил обчислюються окремо, але в правій частині кількох не може бути вказана одна і та ж нечітка змінна. Нечіткі підмножини, призначені для кожної змінної виводу (або одної змінної), об'єднуються разом для формування одної нечіткої підмножини. При подібному об'єднанні зазвичай використовуються операції *max* або *sum*. При композиції максимуму комбінований висновок нечіткої підмножини

конструюється як поточковий максимум по всій нечіткій підмножини (нечітка логіка «АБО»). При композиції суми комбінований вивід нечіткої підмножини конструюється як поточкова сума по всіх нечітких підмножинах, призначених змінній виведення правилами логічного виводу.

4. **Приведення до чіткості** (дефазифікація): цей прийом використовується, коли необхідно перейти від нечіткого виводу до чіткого вихідного значення. При переході від нечіткого виводу до чіткого виходу використовуються різні методи, зокрема, метод центру тяжіння, метод середнього максимуму.

3.6. Основні етапи нечіткого виводу

Розробка і застосування систем нечіткого виводу включає в себе ряд етапів, реалізація яких виконується за допомогою основних положень нечіткої логіки. Інформацією, яка надходить на вхід системи нечіткого виводу, є виміряні деяким чином вхідні змінні. Ці змінні відповідають реальним змінним процесу управління. Інформація, яка формується на виході системи нечіткого виведення, відповідає вихідним змінним, якими є керуючі змінні процесу управління.

Системи нечіткого виводу призначені для перетворення значень вхідних змінних процесу управління у вихідні змінні на основі використання нечітких правил продукцій. Для цього системи нечіткого виводу повинні містити базу правил нечітких продукцій реалізувати нечіткий вивід виводів на основі посилань або умов, представлених у формі нечітких лінгвістичних висловлювань.

Основними етапами нечіткого виводу є (рис.3.34):

- формування бази правил систем нечіткого виводу;
- фазифікація вхідних змінних;
- агрегування підумов у нечітких правилах продукцій;

- активізація або композиція під висновків у нечітких правилах продукції;
- акумулювання висновків нечітких правил продукції.

Елементи теорії нечітких множин, правила імплікації і нечітких міркувань утворюють систему нечіткого виводу. В ній можна виділити множину використовуваних у системі нечітких правил, базу даних, яка містить описи функцій приналежності, а також механізм виведення та агрегування, який формується застосовуваними правилами імплікації. Слід згадати, що у разі технічної реалізації в якості вхідних і вихідних сигналів виступають вимірювані величини, однозначно визначають відповідні вхідним значенням відповідні вихідні значення.

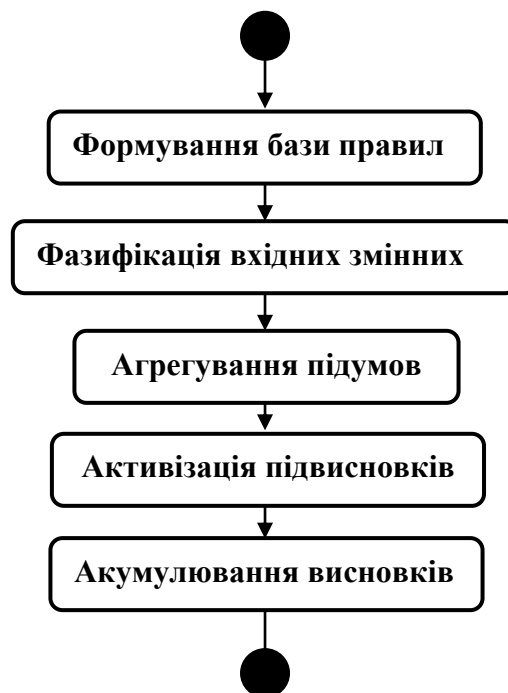


Рис.3.34. Діаграма діяльності процесу нечіткого виводу

Для забезпечення взаємодії множин цих двох видів вводиться нечітка система з так званим **фазифікатором** (перетворювачем множини вхідних даних у нечітку множину) на вході і **дефазифікатором** (перетворювачем нечітких множин в конкретне значення вихідної змінної) на виході. У контексті нечіткої логіки під фазифікацією розуміється не тільки окремий

етап виконання нечіткого виведення, але і власне процес або процедура знаходження значень функцій приналежності нечітких множин (термів) на основі звичайних (не нечітких) вихідних даних. Метою етапу фазифікації є встановлення відповідності між конкретним (зазвичай – чисельним) значенням окремої вхідної змінної системи нечіткого виводу і значенням функції приналежності відповідного їй терма вхідної лінгвістичної змінної. Після завершення цього етапу для всіх вхідних змінних повинні бути визначені конкретні значення функцій приналежності по кожному з лінгвістичних термів, які використовуються в підумовах бази правил системи нечіткого виводу. Структура такої системи представлена на рис.3.35.

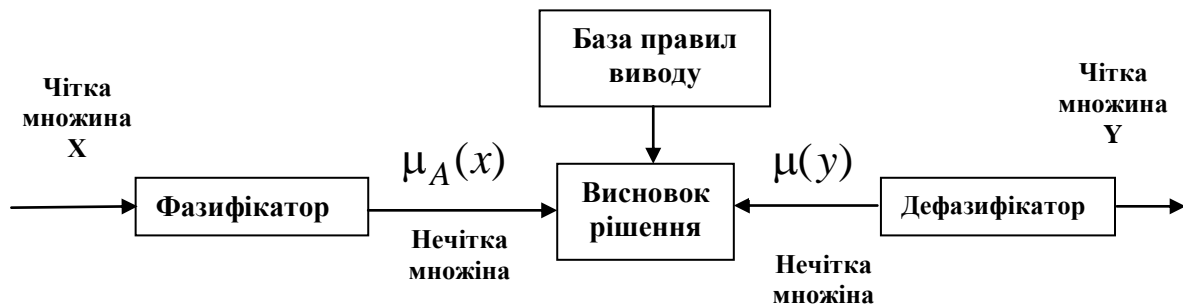


Рис. 3.35. Структура нечіткої системи з фазифікатором і дефазифікатором

Фазифікатор перетворює точну множину вхідних даних на нечітку множину, що визначається за допомогою значень функцій належності, тоді як дефазифікатор вирішує зворотню задачу – він формує однозначне рішення щодо значення вихідної змінної на підставі багатьох нечітких виводів, що виробляються виконавчим механізмом нечіткої системи.

Вихідний сигнал цього модуля може мати M нечітких множин, що визначають діапазон зміни вихідної змінної. Дефазифікатор перетворює цей діапазон в одне конкретне значення, що приймається в якості вихідного сигналу всієї системи. Зазначимо, що існують системи нечіткого виведення, в

яких виконавчий механізм безпосередньо генерує чіткі значення, які вже не потрібно піддавати дефазифікації. Такою системою є система Тагаки-Сугено-Канга.

Узагальнена функціональна структура системи, наведена на рис.3.35., може бути представлена в розширеній формі, яка в явному вигляді демонструє правила нечіткого висновку так, як це зображено на рис. 3.36.

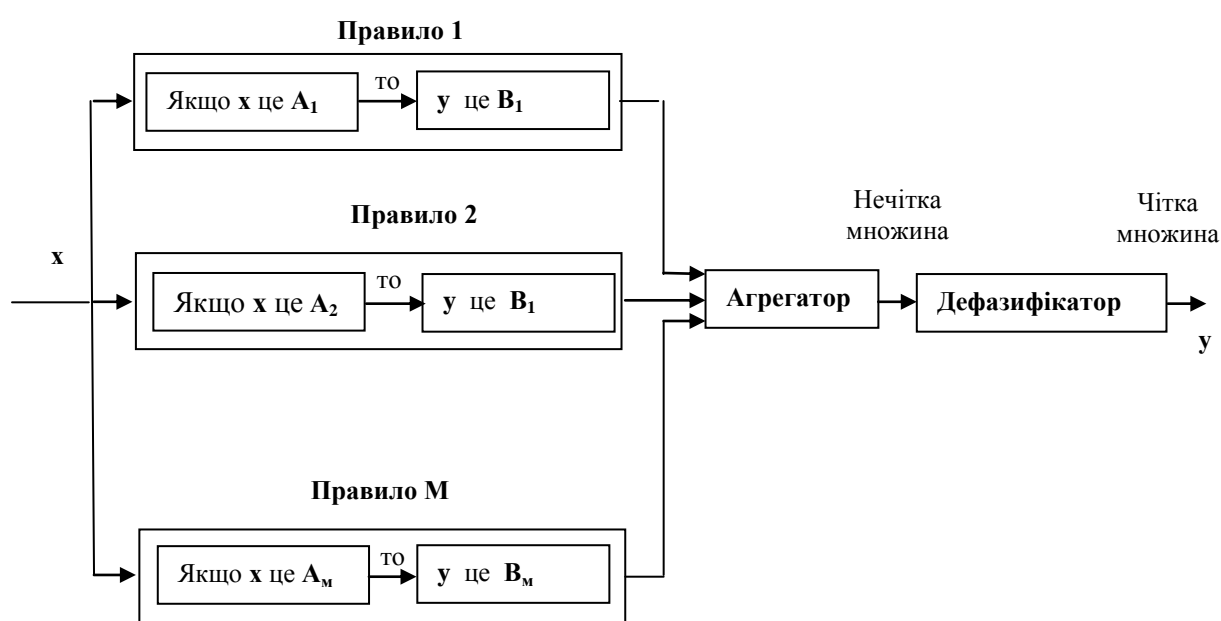


Рис. 3.36. Організація виводу в нечіткій системі при наявності M правил виводу

Оскільки допускається застосування множини нечітких правил, в ній також передбачений блок агрегування, найчастіше реалізується у вигляді логічного суматора (оператор \max). Описувана система виводу називається системою Мамдані-Заде. Вона дуже популярна у звичайних (неадаптивних) нечітких системах.

Як правило, в моделі Мамдані-Заде присутні такі оператори:

- оператор логічного або арифметичного добутку для визначення вихідного рівня активації, в якому враховуються всі компоненти вектора x умови;

- оператор логічного або арифметичного добутку для визначення значення функції приналежності для всієї імплікації $A \rightarrow B$;
- оператор логічної суми як агрегатор рівнозначних результатів імплікації багатьох правил;
- оператор дефазифікації, трансформуючий нечіткий результат $\mu(y)$ в чітке значення вихідної змінної y .

3.6.1. Формування бази правил систем нечіткого виводу

База правил систем нечіткого виведення призначена для формального представлення емпіричних знань або знань експертів у тій чи іншій проблемній області. В системах нечіткого виводу використовуються правила нечітких продукцій, в яких умови та висновки сформульовані в термінах нечітких лінгвістичних висловлювань. Сукупність таких правил називають базами правил нечітких продукцій.

Найпростіший варіант правила нечіткої продукції, який найбільш часто зустрічається в системах нечіткого виведення, може бути записаний у формі:

ПРАВИЛО #: ЯКЩО " $\beta_1 \in \alpha'$ ", ТО " $\beta_2 \in \alpha''$ ".

Тут нечітке висловлювання " $\beta_1 \in \alpha'$ " являє собою умову даного правила нечіткої продукції, а нечітке вислів " $\beta_2 \in \alpha''$ " – нечітке укладення даного правила.

База правил нечітких продукцій являє собою кінцеву множину правил нечітких продукцій, узгоджених щодо використовуваних у них лінгвістичних змінних. Найбільш часто база правил подається у формі структурованого тексту:

ПРАВИЛО 1: ЯКЩО «Умова 1» «Висновок 1» (F_1)

ПРАВИЛО 2: ЯКЩО «Умова 2» «Висновок 2» (F_2)

...

ПРАВИЛО : ЯКЩО «Умова 1» «Висновок » (F_n).

Тут через $F_i (i \in \{1, 2, \dots, n\})$ позначено коефіцієнти визначеності або вагові коефіцієнти відповідних правил. Ці коефіцієнти можуть приймати значення з інтервалу $[0, 1]$, якщо ці вагові коефіцієнти відсутні, зручно прийняти, що їх значення дорівнюють 1.

В якості умов і висновків правил можуть використовуватися тільки нечіткі лінгвістичні висловлювання, при цьому в кожному з нечітких висловлювань мають бути визначені функції належності значень терм-множини для кожної з лінгвістичних змінних.

У системах нечіткого виведення лінгвістичні змінні, які використовуються в нечітких висловлюваннях підумов правил нечітких продукцій, часто називають *вхідними лінгвістичними змінними*, а змінні, які використовуються в нечітких висловлюваннях півисновків правил нечітких продукцій, часто називають *вихідними лінгвістичними змінними*.

Лінгвістична змінна вважається заданою або визначеною, якщо для неї визначена базова терм-множина з відповідними функціями належності кожного терма, а також дві процедури G і M. Найбільш поширеним випадком є використання в якості функцій належності термів трикутних або трапецієподібних функцій належності, розглянутих нижче.

3.6.2. Фазифікація

Під фазифікацією розуміють не тільки окремий етап виконання нечіткого виведення, але і власне процес або процедуру знаходження значень функцій належності нечітких множин (термів) на основі звичайних вихідних даних.

Метою етапу фазифікації є встановлення відповідності між конкретним (зазвичай чисельним) значенням окремої вхідної змінної системи нечіткого виводу і значенням функції належності відповідного їй терма вхідної лінгвістичної змінної. Після

завершення цього етапу для всіх вихідних змінних повинні бути визначені конкретні значення функцій приналежності по кожному з лінгвістичних термів, які використовуються в підумовах бази правил системи нечіткого виводу. До початку етапу фазифікації передбачаються відомими конкретні значення всіх вхідних змінних системи нечіткого виводу.

Далі розглядається кожна з підумов виду « $\beta_i \in \alpha'$ » правил системи нечіткого виводу, де α' – деякий терм з відомою функцією належності $\mu(x)$. При цьому значення використовується як аргумент a_i , тим самим знаходиться кількісне значення. Це значення і є результатом фазифікації підумови « $\beta_i \in \alpha'$ ».

Етап фазифікації вважається закінченим, коли будуть знайдені всі значення $b_i' = \mu(a_i)$ для кожної з підумов всіх правил, що входять до розглянутої бази правил системи нечіткого виводу. При цьому якщо певний терм α'' лінгвістичної змінної β_i не присутній ні в одному з нечітких висловлювань, то відповідне йому значення функції належності не перебуває в процесі фазифікації.

Фазифікатор перетворює мірний вхідний вектор x у нечітку множину A , що характеризується функцією належності $\mu_A(x)$ з чіткими змінними. Незважаючи на те, що нечіткі системи можуть мати функції належності довільної структури, з практичної точки зору найбільшою популярністю користуються функції гауссівського типу, а також трикутні і трапецієподібні функції.

Найбільш характерним прикладом кусково-лінійних функцій належності є "трикутна" (рис.3.11, а) і "трапецієвидна" (рис.3.11, б) функції належності.

Приклад 3.11. В якості прикладу розглянемо ситуацію зі швидкістю руху автомобільного транспорту в межах міста. Хоча

правила дорожнього руху регламентують величину цієї швидкості, проте багато автолюбителів воліють давати власну суб'єктивну оцінку своєї швидкості руху. При цьому використовуються такі визначення, як "мала швидкість", "середня швидкість" і "висока швидкість" руху. Очевидно, що подібна оцінка швидкості може ставитися до діапазону швидкостей в межах інтервалу від 0 км/год до деякої величини, що визначається особистими перевагами того чи іншого водія. Нехай у нашому прикладі з міркування зручності це буде величина 100 км/год.

Формалізація суб'єктивної оцінки швидкості руху може бути виконана за допомогою такої лінгвістичної змінної $\langle \beta_1, T, X, G, M \rangle$, де β_1 – швидкість руху автомобіля; $T = \{ \text{"мала швидкість"}, \text{"середня швидкість"}, \text{"висока швидкість"} \}$; $X = [0, 100]$; G – процедура утворення нових термів за допомогою логічних зв'язок "І", "АБО" і модифікація типу "дуже", "НЕ", "злегка" та ін.. Наприклад: "мала або середня швидкість", "дуже висока швидкість" та ін.; M – процедура завдання на $X = [0, 100]$ нечітких змінних "мала швидкість", "середня швидкість", "висока швидкість", а також відповідних нечітких множин для термів з $G(T)$ відповідно до правил трансляції нечітких зв'язок і модифікаторів "І", "АБО", "НЕ", "дуже", "злегка".

Для розглянутого прикладу нечіткі множини відповідні нечітким змінним: $a_1 = \text{"мала швидкість"}$, $a_2 = \text{"середня швидкість"}$, $a_3 = \text{"висока швидкість"}$, зручно задати графічно за допомогою кусково-лінійних функцій приналежності. Один з можливих варіантів цих нечітких множин зображено на рис.3.37.

Прокоментуємо етап фазифікації трьох нечітких висловлювань на цьому прикладі. Їм відповідають нечіткі висловлювання першого виду: " $\beta_1 \in a_1$ ", " $\beta_1 \in a_2$ ", " $\beta_1 \in a_3$ ". Припустимо, що поточна швидкість автомобіля дорівнює 55 км/год, тобто км/год.

Фазифікація першого нечіткого висловлювання дає в результаті число 0, що означає ступінь істинності і виходить підстановкою значення км/год в якості аргументу функції приналежності терма a_1 (рис.3.37, а). Фазифікація другого нечіткого висловлювання дає в результаті число 0,67 (наближене значення), яке означає його ступінь істинності і виходить підстановкою значення $a_1 = 55$ км/год в якості аргументу функції приналежності терма a_2 (рис. 3.37, б).

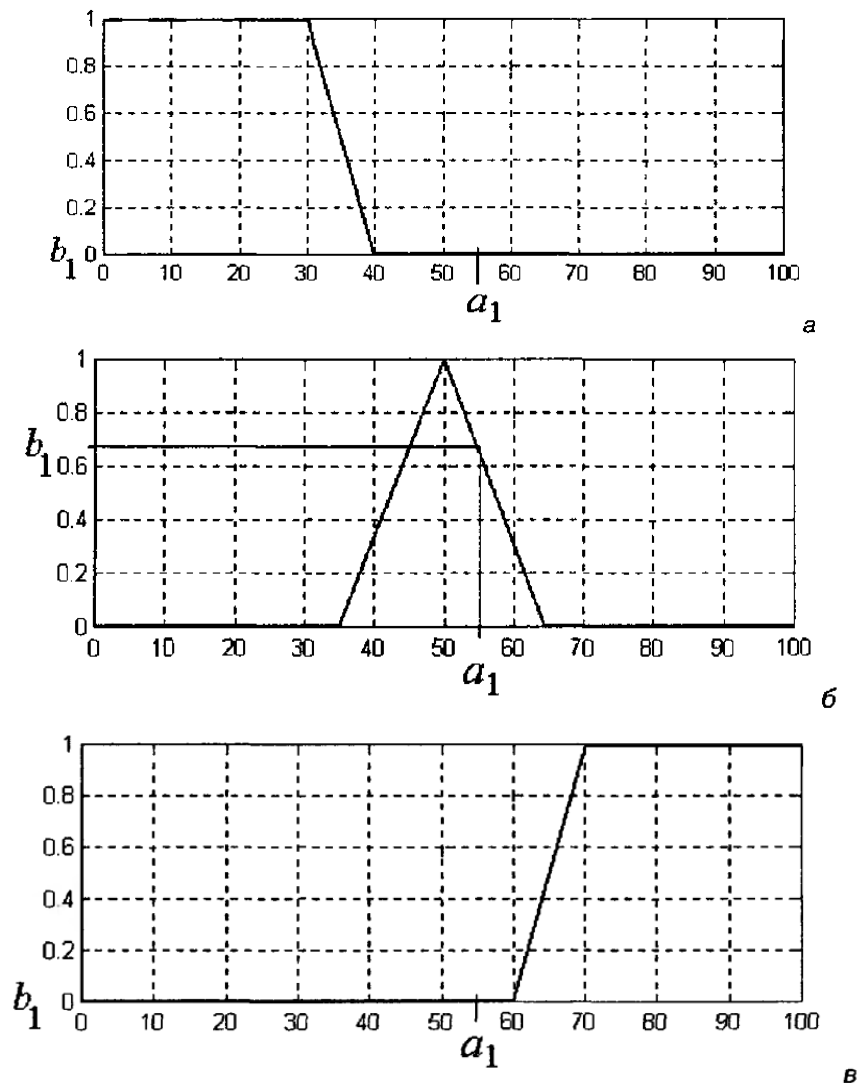


Рис.3.37. Приклад фазифікації вхідних лінгвістичної змінної "швидкість автомобіля" для трьох нечітких висловлювань

Фазифікація третього нечіткого висловлювання дає в результаті число 0, що означає ступінь істинності і виходить

підстановкою значення $a_1 = 55$ км/год в якості аргументу функції належності терма a_3 (рис. 3.37, в).

3.6.3. Агрегування

Агрегування являє собою процедуру визначення ступеня істинності умов по кожному з правил системи нечіткого виводу. Процедура агрегування виконується таким чином. До початку цього етапу передбачаються відомими значення істинності всіх підумов системи нечіткого виводу. Якщо умова правила є нечіткий вислів виду 1 або 2, то ступінь його істинності дорівнює відповідному значенню b_i' .

Якщо ж умова складається з декількох підумов виду:

ПРАВИЛО<#>: ЯКЩО " $\beta_1 \in \alpha'$ " І " $\beta_2 \in \alpha''$ " ТО " $\beta_3 \in v$ "
або

ПРАВИЛО<#>: ЯКЩО " $\beta_1 \in \alpha'$ " АБО " $\beta_2 \in \alpha''$ " ТО " $\beta_3 \in v$ "

причому лінгвістичні змінні в підумовах попарно нерівні одна одній, то визначається ступінь істинності складного висловлювання на основі відомих значень істинності підумов. При цьому для визначення результату нечіткої кон'юнкції або зв'язки «І» може бути використана одна з формул:

логічна кон'юнкція:

$$T(A \wedge B) = \min\{T(a), T(b)\}, \quad (3.3)$$

Алгебраїчний добуток ступенів істинності нечітких висловлювань:

$$T(A \wedge B) = T(a) \cdot T(b), \quad (3.4)$$

граничний добуток ступенів істинності нечітких висловлювань:

$$T(A \wedge B) = \max\{T(a) + T(b) - 1, 0\}, \quad (3.5)$$

драстичний добуток ступенів істинності нечітких висловлювань:

$$T(A \wedge B) = \begin{cases} T(B), & \text{якщо } T(A)=1; \\ T(A), & \text{якщо } T(B)=1; \\ 0, & \text{в інших випадках.} \end{cases} \quad (3.6)$$

Для визначення результату нечіткої диз'юнкції або зв'язки «АБО» може бути використана одна з формул:

логічна диз'юнкція:

$$T(A \vee B) = \max\{T(a), T(b)\}, \quad (3.7)$$

алгебраїчна сума ступенів істинності нечітких висловлювань:

$$T(A \vee B) = T(a) + T(b) - T(A) \cdot T(B), \quad (3.8)$$

гранична сума ступенів істинності нечітких висловлювань:

$$T(A \vee B) = \min\{T(a) + T(b), 1\}, \quad (3.9)$$

драстична сума ступенів істинності нечітких висловлювань:

$$T(A \vee B) = \begin{cases} T(B), & \text{якщо } T(A)=0; \\ T(A), & \text{якщо } T(B)=0; \\ 1, & \text{в інших випадках.} \end{cases} \quad (3.10)$$

При цьому значення b_i' використовуються в якості аргументів відповідних логічних операцій. Тим самим знаходяться кількісні значення істинності всіх умов правил системи нечіткого виводу.

Приклад 3.12. Для ілюстрації виконання цього етапу розглянемо приклад процесу агрегування двох нечітких висловлювань: "швидкість автомобіля середня" І "кави гаряча" і "швидкість автомобіля середня" АБО "кава гаряча" для вхідної лінгвістичної змінної β_1 – швидкість руху автомобіля і β_2 – температура кави. Припустимо, що поточна швидкість автомобіля дорівнює 55 км/год, тобто $a_1 = 55$ км/год, а температура кави дорівнює $a_2 = 70$ °С.

Тоді агрегування першого нечіткого висловлювання з використанням операції нечіткої кон'юнкції (3.5) дає в результаті число $b_1 = 0,67$ (наближене значення), яке означає ступінь істинності і виходить як мінімальне зі значень 0,67 і 0,8 (рис.3.38, а). Агрегування другого нечіткого висловлювання з використанням операції нечіткої диз'юнкції (3.7) дає в результаті число $b_2 = 0,8$, яке означає ступінь істинності і виходить як максимальне із значень 0,67 і 0,8 (рис. 3.38, б).

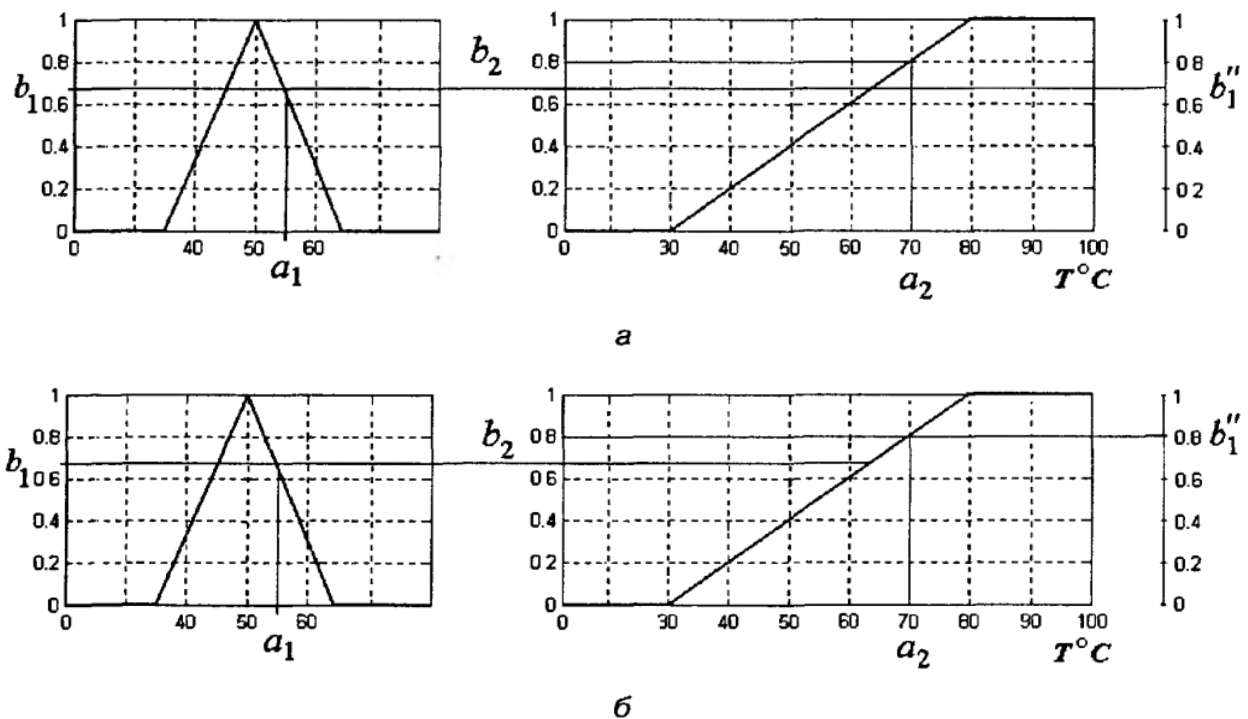


Рис. 3.38. Приклади агрегування підумов для двох нечітких висловлювань "швидкість автомобіля середня" І "висока температура кави" (а) і "швидкість автомобіля середня" АБО "висока температура кави" (б)

Етап агрегування вважається закінченим, коли будуть знайдені всі значення b_k'' для кожного з правил, що входять до розглянутої бази правил P системи нечіткого виводу.

3.6.4. Активація

Активація в системах нечіткого виводу являє собою процедуру або процес знаходження ступеня істинності кожного з підвисновків правил нечітких продукцій. При формуванні бази правил системи нечіткого виводу задаються вагові коефіцієнти для кожного правила (за замовчуванням передбачається, якщо ваговий коефіцієнт не заданий явно, то його значення дорівнює 1).

Формально процедура активації виконується таким чином. До початку цього етапу передбачаються відомими значення істинності всіх умов системи нечіткого виведення, тобто множина значень $B'' = \{b_1'', b_2'', \dots, b_n''\}$ і значення вагових коефіцієнтів F_i для кожного правила. Далі розглядається кожний із висновків правил системи нечіткого виводу. Якщо висновок правила являє собою нечіткий вислів виду 1 або 2, то ступінь його істинності дорівнює алгебраїчному добутку відповідного коефіцієнта b_1'' на ваговий коефіцієнт F_i .

Якщо ж висновок складається з декількох підвисновків виду ПРАВИЛО <#>: *ЯКЩО* " $\beta_1 \in \alpha'$ " І " $\beta_2 \in \alpha''$ " ТО " $\beta_3 \in v$ " або ПРАВИЛО <#>: *ЯКЩО* " $\beta_1 \in \alpha'$ " АБО " $\beta_2 \in \alpha''$ " ТО " $\beta_3 \in v$ ", причому лінгвістичні змінні в підвисновках попарно не дорівнюють одна одній, то ступінь істинності кожного з них дорівнює алгебраїчному добутку відповідного значення b_1'' на ваговий коефіцієнт F_i . Таким чином, знаходяться всі значення ступенів істинності підвисновків для кожного з правил, що входять в даної бази правил системи нечіткого виводу. Цю множину значень позначимо через $C = \{c_1, c_2, \dots, c_q\}$, де q – загальна кількість підвисновків у базі правил.

Після знаходження множини $C = \{c_1, c_2, \dots, c_q\}$ визначаються функції належності кожного з підвисновків для розглянутих вихідних лінгвістичних змінних. Для цієї мети можна використовувати один з методів, що є модифікацією того чи іншого методу нечіткої композиції:

$$\text{min-активізація: } \mu'(y) = \min\{c_i, \mu(y)\}; \quad (3.11)$$

$$\text{prod-активізація: } \mu'(y) = c_i \cdot \mu(y); \quad (3.12)$$

$$\text{average-активізація: } \mu'(y) = 0.5 \cdot (c_i + \mu(y)),$$

де $\mu(y)$ – функція належності терма, який є значенням деякої вихідної змінної, заданої на універсумі Y .

Приклад 3.13. Для ілюстрації виконання цього етапу розглянемо приклад процесу активізації укладання в наступному правилі нечіткої продукції (це правило навряд чи має цільове застосування і використовується формальним чином): ЯКЩО "середня швидкість автомобіля" ТО "кава гаряча". Вхідною лінгвістичною змінною в цьому правилі є β_1 – швидкість руху автомобіля, а вихідною змінною є β_2 – температура кави. Припустимо, що поточна швидкість автомобіля дорівнює 55 км/год, тобто $a_1 = 55$ км/ч.

Оскільки агрегування умови цього правила дає в результаті $b_1 = 0,67$, а ваговий коефіцієнт дорівнює 1 (за замовчуванням), то значення 0,67 буде використовуватися в якості c_1 для отримання результату активізації. Результат, отриманий методом min-активізації (3.11), зображений (на рис. 3.39, а) більш темним кольором, а результат, отриманий методом prod-активізації (3.12), зображений (на рис. 3.39, б) більш темним кольором.

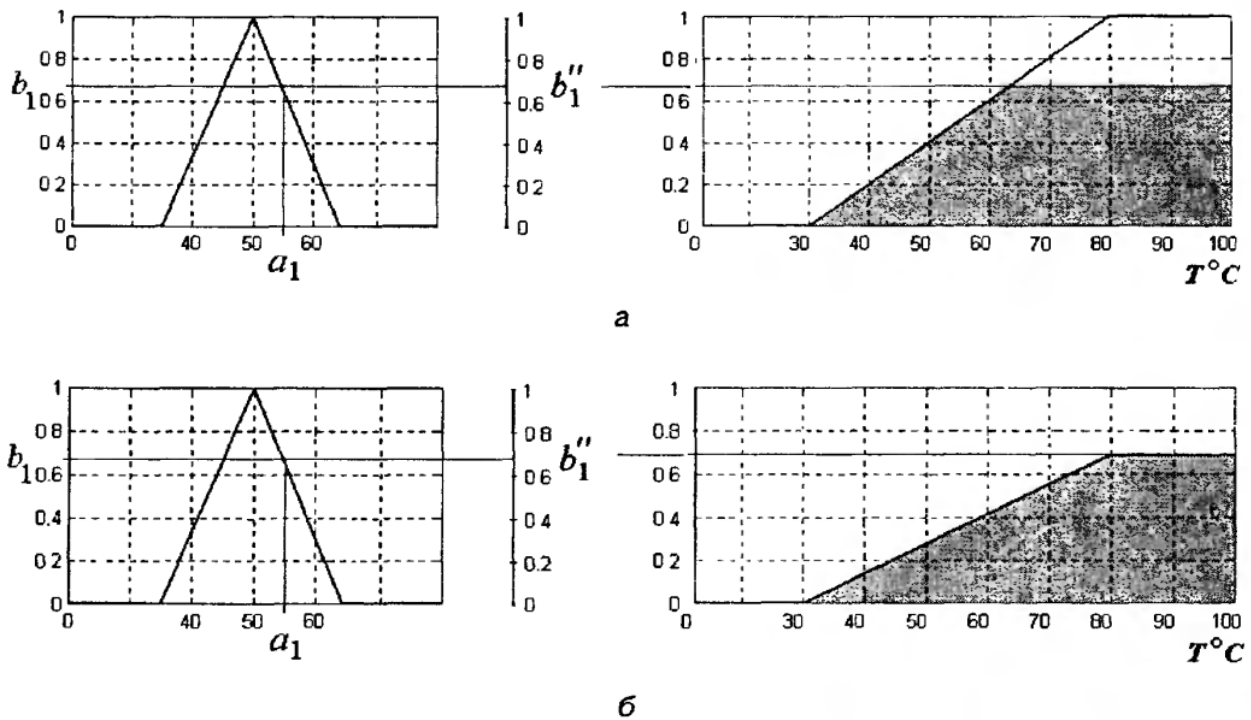


Рис. 3.39. Приклад активізації укладення правила нечіткої продукції

Слід пам'ятати, що в цьому прикладі – на відміну від попереднього "температура кави" – вихідна лінгвістична змінна. Етап активізації вважається закінченим, коли для кожної з вихідних лінгвістичних змінних, що входять в окремі підвисновки правил нечітких продукцій, будуть визначені функції належності нечітких множин значень, тобто сукупність нечітких множин: c_1, c_2, \dots, c_q , де q – загальна кількість підвисновків у базі правил.

3.6.5. Акумуляція

Акумуляція або акумулювання в системах нечіткого виводу являє собою процедуру або процес знаходження функції належності для кожної з лінгвістичних змінних множини $W = \{\omega_1, \omega_2, \dots, \omega_s\}$.

Мета акумуляції полягає в тому, щоб об'єднати або акумулювати всі ступені істинності висновків (підвисновків) для отримання функції належності кожної з вихідних змінних. Причина необхідності виконання цього етапу полягає в тому, що підвисновки, що відносяться до однієї і тієї ж вихідної лінгвістичної змінної, належать різним правил системи нечіткого виводу.

Формально процедура акумуляції виконується таким чином. До початку цього етапу передбачаються відомими значення істинності всіх підвисновків для кожного з правил R_k , входять у розглянуту бази правил системи нечіткого виводу, у формі сукупності нечітких множин: c_1, c_2, \dots, c_q , де q – загальна кількість підвисновків у базі правил. Потім послідовно розглядаються кожна з вихідних лінгвістичних змінних та нечіткі множини, що відносяться до неї. Результат акумуляції для вихідної лінгвістичної змінної визначається як об'єднання нечітких множин за однією з формул: об'єднання нечітких множин, алгебраїчне об'єднання нечітких множин, граничне об'єднання нечітких множин, драстичне об'єднання нечітких множин.

Приклад 3.14. Для ілюстрації виконання цього етапу розглянемо приклад процесу акумуляції висновків для трьох нечітких множин C_{11}, C_{12}, C_{13} , отриманих у результаті виконання процедури активізації для вихідної лінгвістичної змінної "швидкість руху автомобіля" в деякій системі нечіткого виводу. Припустимо, що функції приналежності цих нечітких множин зображені на рис. 3.40, а, б, в відповідно.

Акумуляція цих функцій приналежності методом max-об'єднання нечітких множин C_{11}, C_{12}, C_{13} дозволяє отримати в результаті функцію належності вихідної лінгвістичної змінної "швидкість руху автомобіля", яка представлена на рис. 3.40, г. Ця функція належності відповідає нечіткій множині C_1' , прийнявши, що розглянута вихідна лінгвістична змінна є ω_1 .

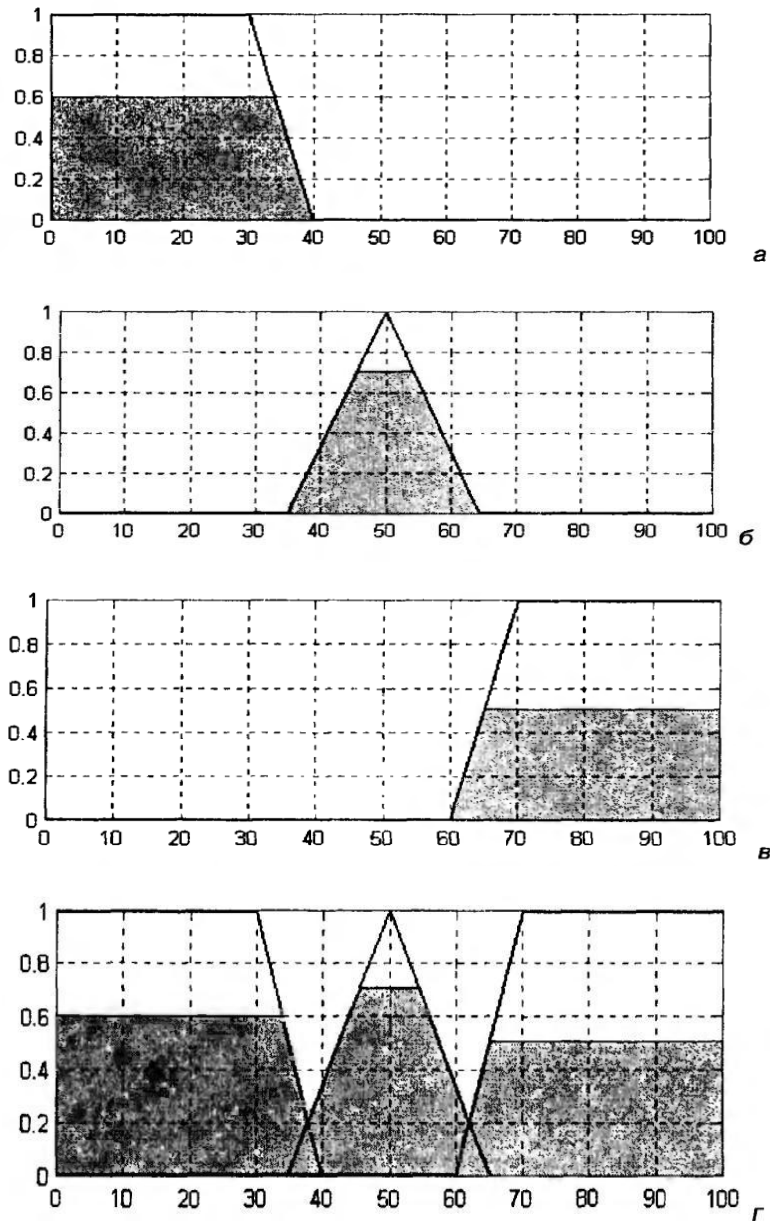


Рис. 3.40. Приклад акумуляції висновків для вихідної лінгвістичної змінної "швидкість руху автомобіля"

Етап акумуляції вважається закінченим, коли для кожної з вихідних лінгвістичних змінних будуть визначені підсумкові функції приналежності нечітких множин їх значень.

3.6.6. Дефазифікація

Дефазифікація в системах нечіткого виведення являє собою процедуру або процес знаходження звичайного значення для кожної з вихідних лінгвістичних змінних.

Мета дефазифікації полягає в тому, щоб, використовуючи результати акумуляції всіх вихідних лінгвістичних змінних, отримати звичайне кількісне значення кожної з вихідних змінних, яке може бути використане спеціальними пристроями, зовнішніми по відношенню до системи нечіткого виводу.

Формально процедуру дефазифікації виконується таким чином. До початку цього етапу передбачаються відомими функції приналежності всіх вихідних лінгвістичних змінних у формі нечітких множин: C'_1, C'_2, \dots, C'_s , де s – загальна кількість вихідних лінгвістичних змінних у бази правил системи нечіткого виводу. Результат дефазифікації для вихідної лінгвістичної змінної визначається у вигляді кількісного значення, одержуваного за однією з розглянутих нижче формул.

Етап дефазифікації вважається закінченим, коли для кожної з вихідних лінгвістичних змінних будуть визначені підсумкові кількісні значення у формі деякого дійсного числа, тобто у вигляді u_1, u_2, \dots, u_s .

Для виконання чисельних розрахунків на етапі дефазифікації можуть бути використані такі формули, які дістали назви методів дефазифікації.

Метод центру ваги

Центр тяжкості або центроїд площі розраховується за формулою:

$$y = \frac{\int_{\min}^{\max} x \cdot \mu(x) dx}{\int_{\min}^{\max} x \cdot \mu(x) dx},$$

де y – результат дефазифікації; x – змінна, що відповідає вихідній лінгвістичній змінній; $\mu(x)$ – функція належності нечіткої множини, відповідній вихідній змінній після етапу акумуляції; a і b – ліва і права точки інтервалу носія нечіткої множини розглянутої вихідної змінної.

При дефазифікації методом центру тяжіння звичайне значення вихідної змінної дорівнює абсцисі центру ваги площі, обмеженої графіком кривої функції належності відповідної вихідної змінної.

Приклад дефазифікації методом центру тяжіння функції належності вихідної лінгвістичної змінної "швидкість руху автомобіля" зображено на рис. 3.41. У цьому випадку $y_1 = 40$ км/год (наближене значення).

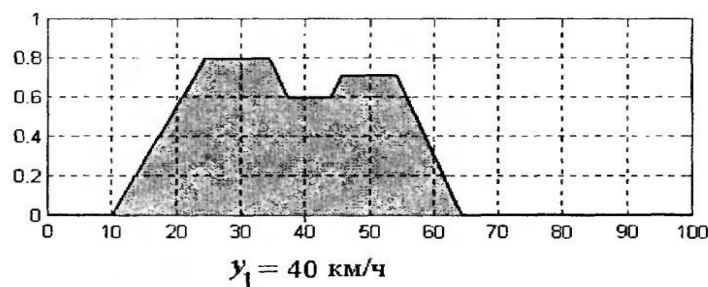


Рис. 3.41. Приклад дефазифікації вихідної лінгвістичної змінної "швидкість руху автомобіля" методом центру тяжіння

Метод центру тяжіння для одноточкових множин

Центр тяжіння для одноточкових множин розраховується за формулою:

$$y = \frac{\sum_{i=1}^n x_i \cdot \mu(x_i)}{\sum_{i=1}^n \mu(x_i)},$$

де n – число одноточкових (одноелементних) нечітких множин, кожна з яких характеризує єдине значення розглянутої вихідної лінгвістичної змінної.

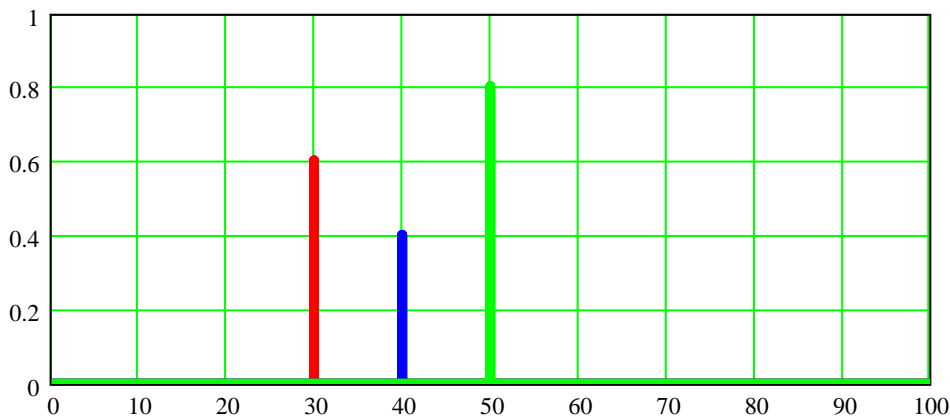


Рис.3.42. Приклад дефазифікації вихідної лінгвістичної змінної «швидкість руху автомобіля» методом центру тяжіння для одноточкових множин

Приклад дефазифікації методом центру тяжіння для одноточкових множин функції належності вихідної лінгвістичної змінної «швидкість руху автомобіля» показаний на рис. 3.42. У цьому випадку $y_1 = 41$ км/год (наближене значення):

$$y_1 = \frac{30 \cdot 0,6 + 40 \cdot 0,4 + 50 \cdot 0,8}{0,6 + 0,4 + 0,8}.$$

Метод центру площі

Центр площі дорівнює $y = u$, де значення визначається з рівняння:

$$\int_{\min}^u \mu(x) dx = \int_u^{\max} \mu(x) dx.$$

Центр площі дорівнює абсцисі, яка ділить площу, обмежену графіком кривої функції належності відповідної вихідної змінної, на дві рівні частини.

Приклад дефазифікації методом центру площі функції належності вихідної лінгвістичної змінної "швидкість руху автомобіля" зображено на рис. 3.40. У цьому випадку $y_1 = 35$ км/год (наближене значення).

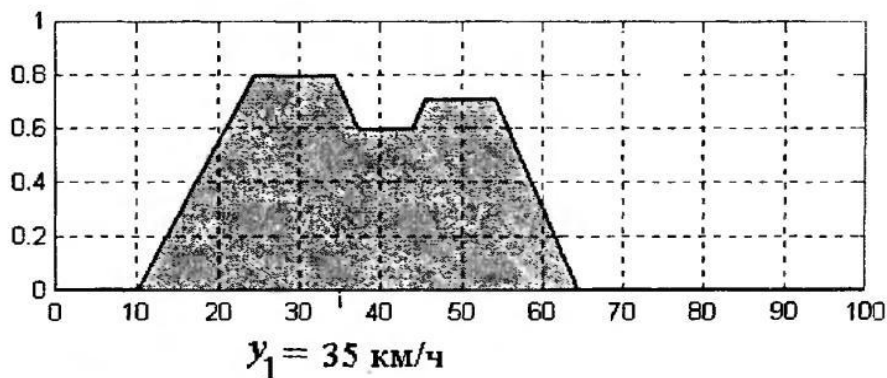


Рис. 3.43. Приклад дефазифікації вихідної лінгвістичної змінної "швидкість руху автомобіля" методом центру площі

3.7. Основні алгоритми нечіткого висновку

Нечіткий алгоритм – впорядкована множина нечітких правил, у формулюванні яких містяться нечіткі вказівки.

Нехай деяка система описується такими нечіткими правилами:

П1: якщо $x \in A$, то $\omega \in D$;

П2: якщо $y \in B$, то $\omega \in E$,

П3: якщо $z \in Z$, то $\omega \in F$,

де x, y і z – імена вхідних змінних; ω – ім'я змінної виводу; A, B, C, D, E, F – деякі задані значення ФП трикутної форми.

Процедура отримання логічного виводу ілюструється рис. 3.44. Передбачається, що входні змінні прийняли деякі конкретні (чіткі) значення x_0, y_0 і z_0 . Відповідно до наведених вище етапів, на першому етапі для даних значень і виходячи з їх функцій належності А, В, С, знаходяться ступені істинності $\alpha(x_0), \alpha(y_0), \alpha(z_0)$ для передумов кожного з трьох наведених правил.

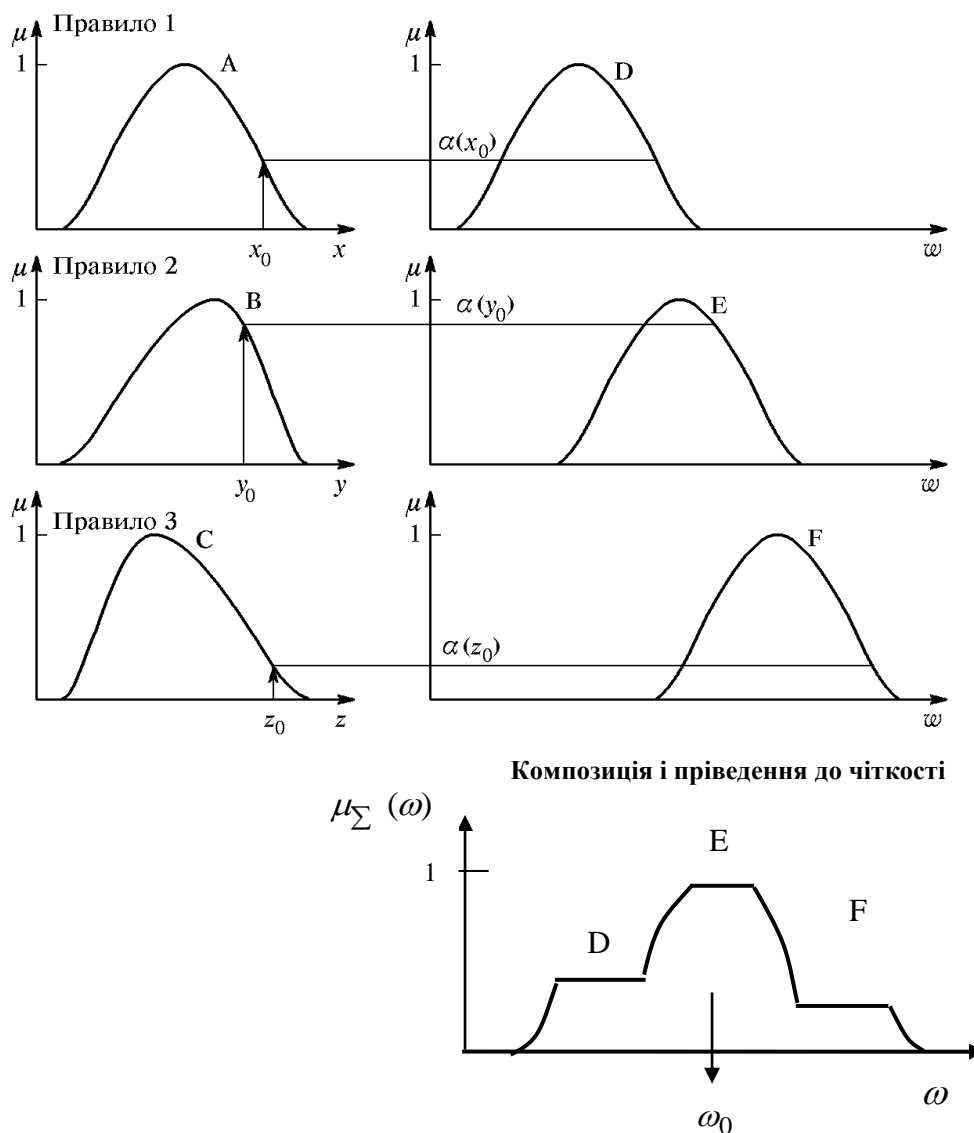


Рис.3.44. Ілюстрація до процедури логічного виводу

На етапі 2 виникає «відсікання» функцій належності висновків правил (тобто D, E, F) на рівнях $\alpha(x_0), \alpha(y_0), \alpha(z_0)$.

На третьому етапі розглядаються усічені на другому етапі функції належності і проводиться їх об'єднання з використанням операції \max , у результаті чого виходить комбінована нечітка підмножина, що описується функцією належності $\mu_{\Sigma}(\omega)$ і відповідна логічному виводу для вихідної змінної ω .

На 4-му етапі – при необхідності – знаходиться чітке значення вихідної змінної, наприклад, із застосуванням центроїдного методу: чітке значення вихідної змінної визначається як центр тяжіння для кривої $\mu_{\Sigma}(\omega)$, тобто

$$\omega_0 = \frac{\int_{\Omega} \omega \mu_{\Sigma}(\omega) d\omega}{\int_{\Omega} \mu_{\Sigma}(\omega) d\omega}.$$

Розглянемо такі найбільш часто використовувані модифікації алгоритму нечіткого виводу, вважаючи, що базу правил утворюють два нечітких правила:

П1: якщо $x \in A_1$ і $y \in B_1$, то $z \in C_1$;

П2: якщо $x \in A_2$ і $y \in B_2$, то $z \in C_2$,

де x і y – імена вхідних змінних; z – ім'я змінної виводу; $A_1, A_2, B_1, B_2, C_1, C_2$ – деякі задані значення ФП, при цьому чітке значення z_0 необхідно визначити на основі наведеної інформації та чітких значень x_0 і y_0 .

Вибір конкретних варіантів параметрів кожного з етапів визначає певний алгоритм, який у повному обсязі реалізує нечіткий висновок у системах правил нечітких продукцій. До теперішнього часу запропоновано декілька алгоритмів нечіткого висновку.

Алгоритм Мамдані (Mamdani)

Алгоритм Мамдані є одним з перших, який знайшов застосування в системах нечіткого виведення. Даний алгоритм математично може бути описаний таким чином.

1. Нечіткість (ФП, визначені на вхідних змінних, застосовуються до їх фактичних значень для визначення ступеня істинності): знаходяться ступені істинності для передумов кожного правила $A_1(x_0), A_2(x_0), B_1(y_0), B_2(y_0)$.

2. Нечіткий вивід: знаходяться рівні «відсікання» для передумов кожного з правил з використанням операції «min»

$$\alpha_1 = A_1(x_0) \wedge B_1(y_0);$$

$$\alpha_2 = A_2(x_0) \wedge B_2(y_0);$$

Після чого знаходяться «усічені» функції належності:

$$C'_1(z) = (\alpha_1 \wedge C_1(z));$$

$$C'_2(z) = (\alpha_2 \wedge C_2(z)).$$

3. Композиція – виходи всіх правил обчислюються окремо, але у правій частині декількох них може бути вказана одна і та ж змінна; нечіткі підмножини, призначені для кожної змінної висновку, об'єднуються разом для формування одної нечіткої підмножини. Проводиться об'єднання знайдених усічених функцій з використанням операції «max», що призводить до отримання підсумкової нечіткої підмножини для змінної виходу з ФП виду:

$$\mu_{\Sigma}(z) = C(z) = C_1(z) \vee C_2(z) = (\alpha_1 \wedge C_1(z)) \vee (\alpha_2 \wedge C_2(z)).$$

4. Приведення до чіткості для відшукування z_0 визначається методом центру тяжіння.

Алгоритм Цукамото

Вихідні посилання ті ж, як і у попереднього алгоритму, але функції є монотонними $C_1(z), C_2(z)$.

1. Перший етап – такий же, як в алгоритмі Мамдані.

2. На другому етапі визначаються рівні відсікання, а потім за допомогою рішення рівняння $\alpha_1 = C_1(z_1)$, $\alpha_2 = C_2(z_2)$ знаходять чіткі значення z_1, z_2 для кожного з вихідних правил.

3. Визначають чітке значення змінної висновку як зважене середнє z_1, z_2 :

$$z_0 = \frac{\alpha_1 z_1 + \alpha_2 z_2}{\alpha_1 + \alpha_2};$$

у загальному випадку (дискретний варіант центроїдного методу)

$$z_0 = \frac{\sum_{i=1}^n \alpha_i z_i}{\sum_{i=1}^n \alpha_i}.$$

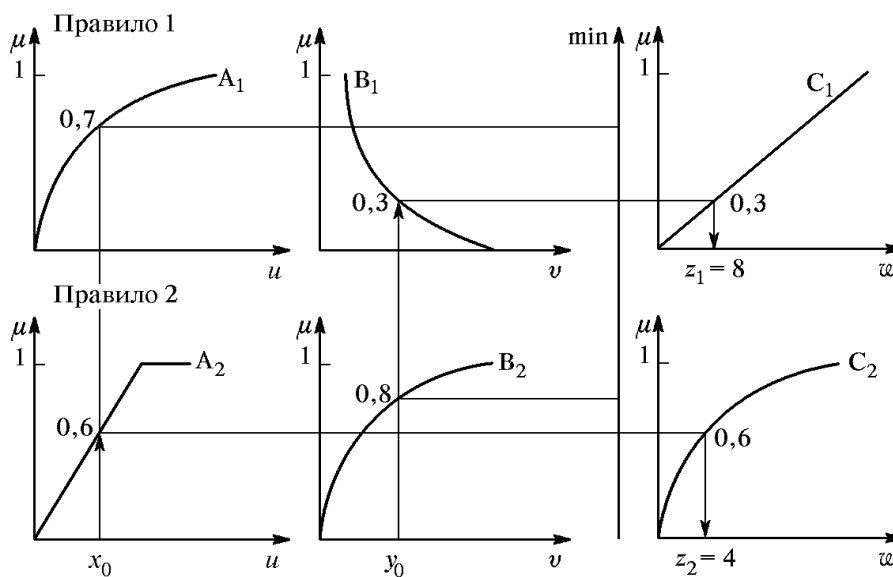


Рис.3.45. Ілюстрація до алгоритму Цукамото

Приклад 3.15. Нехай маємо $A_1(x_0) = 0,7$, $A_2(x_0) = 0,6$, $B_1(y_0) = 0$, $B_2(y_0) = 0,8$. Відповідні рівні відсікання

$$\alpha_1 = \min(A_1(x_0), B_1(y_0)) = \min(0,7; 0,3) = 0,3,$$

$$\alpha_2 = \min(A_2(x_0), B_2(y_0)) = \min(0,6; 0,8) = 0,6$$

і значення $z_1 = 8, z_2 = 4$ знайдені в результаті розв'язання рівнянь

$$C_1(z_1) = 0,3, \quad C_2(z_2) = 0,6.$$

При цьому чітке значення змінної виведення

$$z_0 = (8 \cdot 0,3 + 4 \cdot 0,6) / (0,3 + 0,6) = 6.$$

Алгоритм Сугено

У цьому алгоритмі використано набір правил у такій формі:

П1: якщо $x \in A_1$ і $y \in B_1$, то $z = a_1x + b_1y$;

П2: якщо $x \in A_2$ і $y \in B_2$, то $z = a_2x + b_2y$.

1. Перший етап – такий же, як в алгоритмі Мамдані.
2. На другому етапі визначаються значення

$$\alpha_1 = A_1(x_0) \wedge B_1(y_0);$$

$$\alpha_2 = A_2(x_0) \wedge B_2(y_0)$$

та індивідуальні виходи правил

$$z_1^* = a_1x_0 + b_1y_0;$$

$$z_2^* = a_2x_0 + b_2y_0.$$

Ілюстрація алгоритму на рис. 3.46.

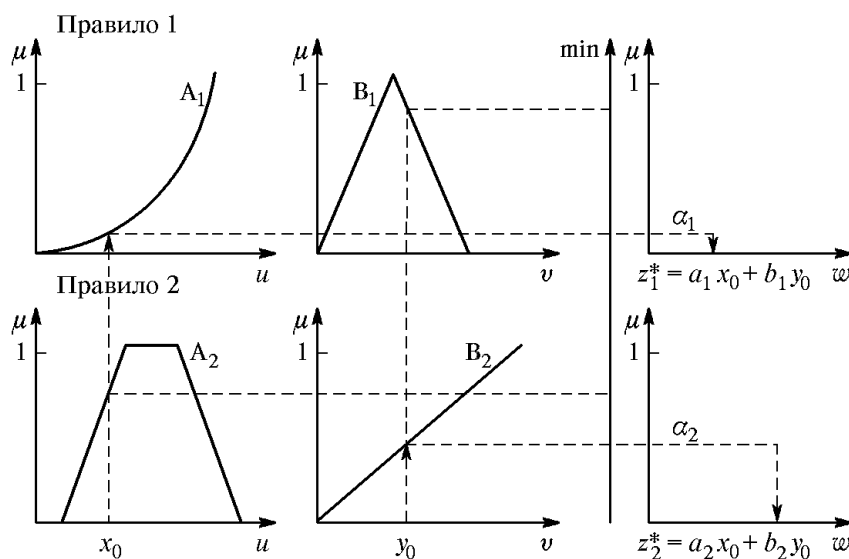


Рис. 3.46. Ілюстрація до алгоритму Сугено

3. На третьому етапі розраховується чітке значення змінної:

$$z_0 = \frac{\alpha_1 z_1^* + \alpha_2 z_2^*}{\alpha_1 + \alpha_2}.$$

Алгоритм Ларсена

1. Перший етап – такий же, як в алгоритмі Мамдані.

2. На другому етапі визначаються значення

$$\alpha_1 = A_1(x_0) \wedge B_1(y_0);$$

$$\alpha_2 = A_2(x_0) \wedge B_2(y_0)$$

а потім приватні нечіткі підмножини $\alpha_1 C_1(z), \alpha_2 C_2(z)$.

3. Знаходять підсумкове нечітка підмножина з ФП виду:

$$\mu_{\Sigma}(z) = C(z) = \alpha_1 C_1(z) \vee \alpha_2 \wedge C_2(z).$$

Приведення до чіткості як і в попередніх випадках.

Алгоритм Ларсена ілюструється на рис. 3.47.

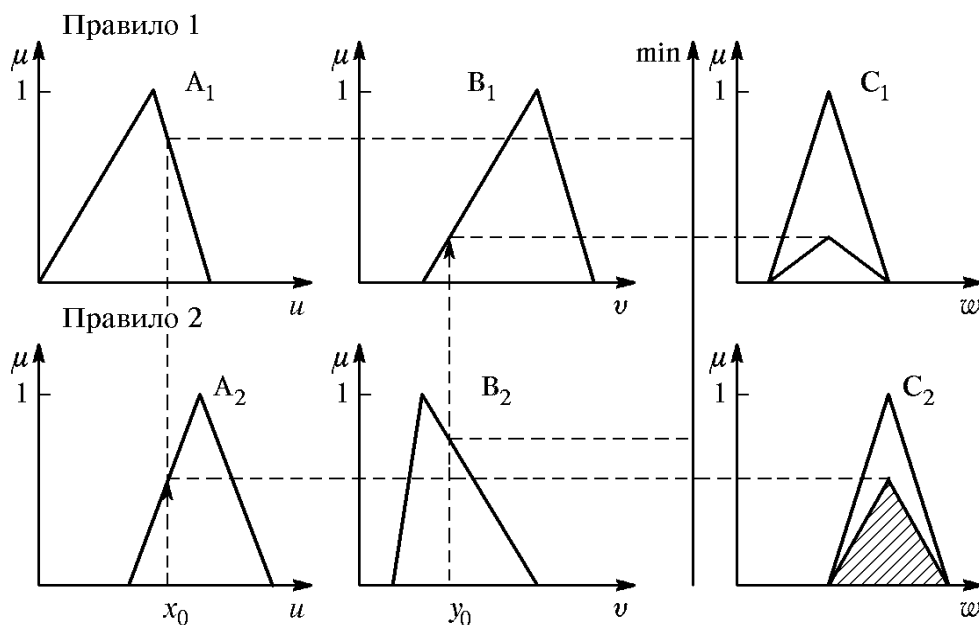


Рис.3.47. Ілюстрація до алгоритму Ларсена

Спрощений алгоритм нечіткого висновку. У цьому алгоритмі використано набір правил у такій формі:

$$\text{П1: якщо } x \in A_1 \text{ і } y \in B_1, \text{ то } z = C_1;$$

П2: якщо $x \in A_2$ і $y \in B_2$, то $z = C_2$.

C_1, C_2 – деякі чіткі числа.

1. Перший етап – такий же, як в алгоритмі Мамдані.
2. На другому етапі визначаються значення

$$\alpha_1 = A_1(x_0) \wedge B_1(y_0);$$

$$\alpha_2 = A_2(x_0) \wedge B_2(y_0).$$

3. Визначають чітке значення змінної висновку як зважене середнє:

$$z_0 = \frac{\alpha_1 C_1 + \alpha_2 C_2}{\alpha_1 + \alpha_2}.$$

Ілюстрація алгоритму наведена на рис. 3.48.

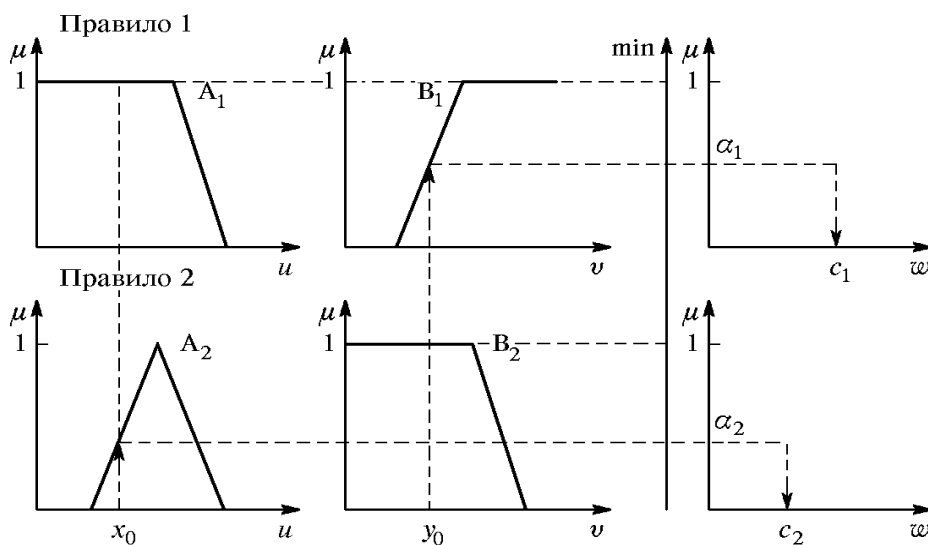


Рис.3.48. Ілюстрація спрощеного алгоритму нечіткого висновку

Нечіткий вивід

Нечітка імплікація. Нечіткою імплікацією чи просто імплікацією нечітких висловлювань A і B (записується як: $A \supset B$ та читається "з A слідує B ", "ЯКЩО A , ТО B ") називається бінарна логічна операція, результат якої є нечітким висловлюванням, істинність якого може набувати значення, обумовлене однією з наступних формул.

- Класична нечітка імплікація, запропонована Л. Заде:

$$T(A \supset B) = \max\{\min\{T(A), T(B)\}, 1 - T(A)\}$$

Цю форму нечіткої імплікації називають також нечіткою імплікацією Заде.

- Класична нечітка імплікація для випадку $T(A) \geq T(B)$:

$$T(A \supset B) = \max\{T(\bar{A}), T(B)\} = \max\{1 - T(A), T(B)\} \text{ або}$$

$$\mu_{A \rightarrow B} = \max\{1 - \mu_A(x), \mu_B(x)\}$$

- Нечітка імплікація запропонована Е. Мамдані:

$$T(A \supset B) = \min\{T(A), T(B)\}$$

Основою для проведення операції нечіткого логічного висновку є база правил, що містить нечіткі висловлювання у формі "Якщо то" і функції приналежності для відповідних лінгвістичних термів. При цьому повинні дотримуватися такі умови:

- існує хоча б одне правило для кожного лінгвістичного терма вихідної змінної;
- для будь-якого терма вхідної змінної є хоча б одне правило, в якому цей терм використовується як передумова (ліва частина правила).

В іншому випадку має місце неповна база нечітких правил. Нечіткими висловлюваннями називаються:

1. Висловлювання $\langle \beta \in \beta' \rangle$, де β – найменування лінгвістичної змінної, β' – її значення, якому відповідає нечітка множина на універсальній множині X .

2. Висловлювання $\langle \beta \in m\beta' \rangle$, де m – модифікатор, якому відповідають слова «дуже», «більш-менш», «дуже багато» та ін.

3. Складові висловлювання, утворені з висловлювань видів 1 і 2 і сполучників "І", "АБО", "ЯКЩО .., ТО ...", "ЯКЩО .., ТО .., ІНАКШЕ".

Приклад розв'язання задачі

Приклад 3.16. Побудувати нечітку базу знань (використовувати не менше 3 лінгвістичних змінних) для задачі визначення тимчасових витрат для розв'язання студентом завдань даного посібника (ураховувати успішність студента і кількість варіантів, що виконуються), перевірити її на повноту і призвести нечіткий вивід для конкретних значень (вибрати випадковим способом).

Опис процесу розв'язання. Для реалізації логічного висновку необхідно виконати таке:

1. Сформулювати природною мовою у вигляді пропозицій «Якщо ..., то» закономірності предметної області.

2. Виділити з цих речень лінгвістичні змінні, їх значення (побудувати їх функції приналежності), висловлювання різних видів, формалізувати нечіткі правила.

3. Перевірити отриману базу знань на повноту.

4. Провести фазифікацію (вхідні дані обираємо випадковим чином).

5. Провести акумуляцію.

6. Провести дефазифікацію.

Розв'язання.

1. Речення, що описують задачу, такі:

- *Якщо успішність студента висока або добра і він розв'язує малу кількість варіантів, то йому потрібно небагато часу.*
- *Якщо успішність студента висока або добра і він розв'язує багато варіантів, то йому потрібно досить великий проміжок часу.*
- *Якщо успішність студента низька і він розв'язує багато варіантів, то йому потрібно багато часу.*
- *Якщо успішність студента середня і він розв'язує достатньо велику кількість варіантів, то йому потрібно досить великий проміжок часу.*

Виділити з цих речень лінгвістичні змінні (визначають їх через формальний запис $\langle \beta, T, X, G, M \rangle$):

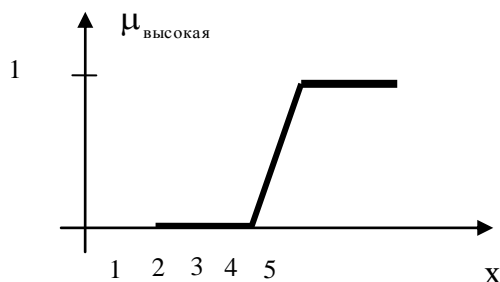
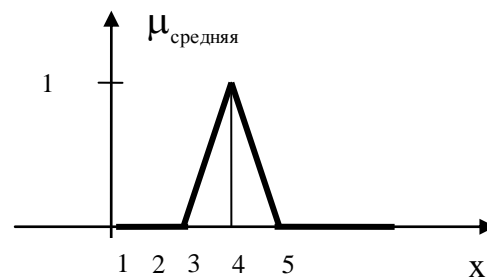
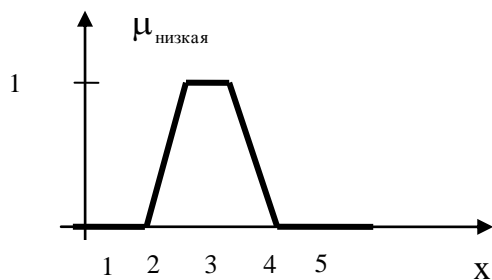
1. β = успішність студента, T =(«висока», «середня», «низька»), $X=[2,5]$ (використовується п'ятибальна система), G =(«дуже низька», «висока чи середня»), M – зменшення на одиницю ступеня належності нечіткої змінної «висока», операція об'єднання нечітких множин;

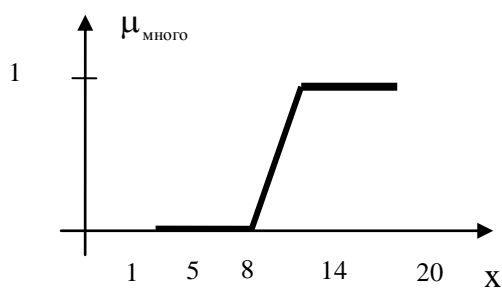
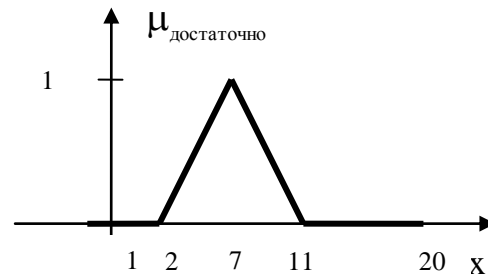
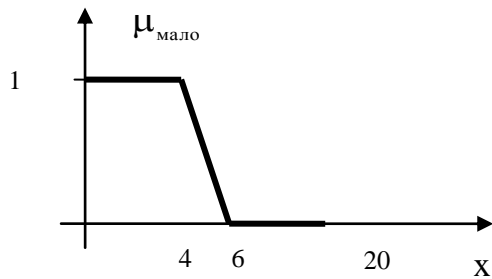
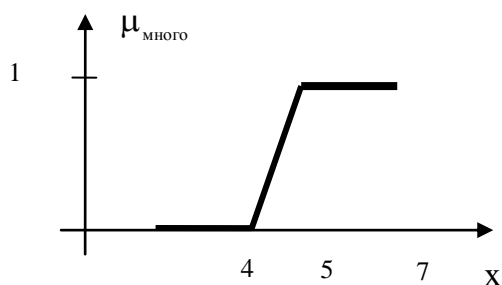
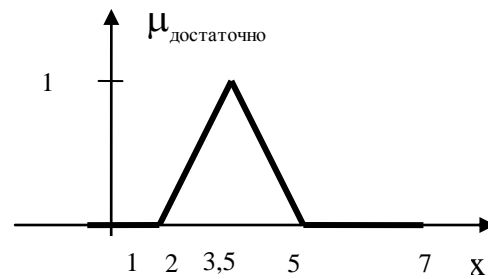
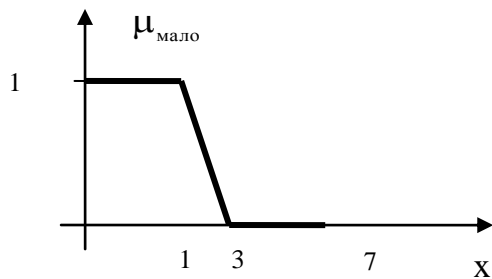
2. β = кількість варіантів, T =(«мало», «достатньо», «багато»), $X=[1,20]$ (кількість варіантів 20 у кожній темі), G =(«дуже багато», «достатньо чи мало»), M – збільшення на одиницю ступеня приналежності нечіткої змінної «багато», операція об'єднання нечітких множин;

3. β = кількість часу, T =(«мало», «достатньо», «багато»), $X=[1,7]$ (кількість годин на тиждень, відданих предмету вивчення), G =(«дуже багато», «достатньо чи мало»), M – збільшення на одиницю ступеня приналежності нечіткої змінної «багато», операція об'єднання нечітких множин.

Для повного завдання лінгвістичної змінної необхідно визначити нечіткі змінні, що входять у T :

Успішність:



Кількість варіантів:**Кількість часу:**

З урахуванням виділених лінгвістичних змінних, нечіткі правила такі:

1. Якщо Успішність = «висока» або Успішність = «середня» і Кількість варіантів = «мало», то Кількість часу = «мало».

2. Якщо Успішність = «висока» або Успішність = «середня» і Кількість варіантів = «багато», то Кількість часу = «досить».

3. Якщо Успішність = «низька» і Кількість варіантів = «багато», то Кількість часу = «багато».

4. Якщо Успішність = «середня» і Кількість варіантів = «достатньо», то Кількість часу = «досить».

2. Перевіримо отриману базу на повноту:

- існує хоча б одне правило для кожного лінгвістичного терма вихідної змінної – вихідна змінна «Кількість часу» має 3 терма: «мало» використовується в 1 правилі, «достатньо» – в 2 і 4, «багато» – у третьому;

- для будь-якого терма вхідної змінної є хоча б одне правило, в якому цей терм використовується як передумови – є дві вхідних змінних «Успішність» і «Кількість варіантів» у кожній з них 3 терма: «висока» використовується в 1 і 2 правилах, «середня» – в 1, 2 і 4, «низька» – в 3, «мало» – в 1, «достатньо» – 4, «багато» – 3 і 2.

Значить, отримана база нечітких правил повна.

3. Нехай є студент Іванов А.А., що має середню оцінку 3,5 і вирішив розв'язати 9 варіантів, потрібно визначити скільки йому знадобиться часу.

Визначимо ступені впевненості найпростіших тверджень:

Успішність = «висока» – 0;

Успішність = «середня» – 0.5;

Успішність = «низька» – 1;

Кількість варіантів = «мало» – 0;

Кількість варіантів = «достатньо» – 0.5;

Кількість варіантів = «багато» – 0.125.

Визначимо ступені впевненості посилок правил:

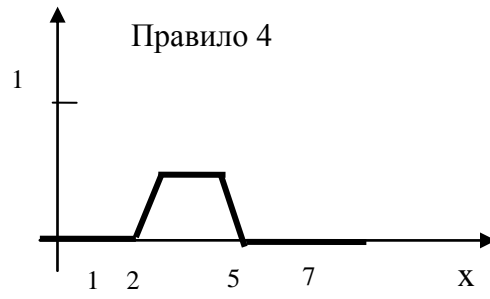
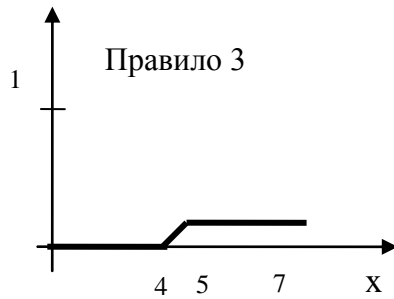
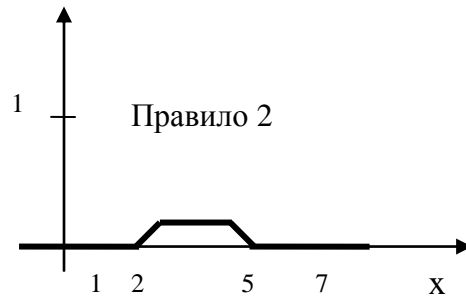
Правило 1: $\min(\max(0, 0.5), 0) = 0$;

Правило 2: $\min(\max(0, 0.5), 0.125) = 0.125$;

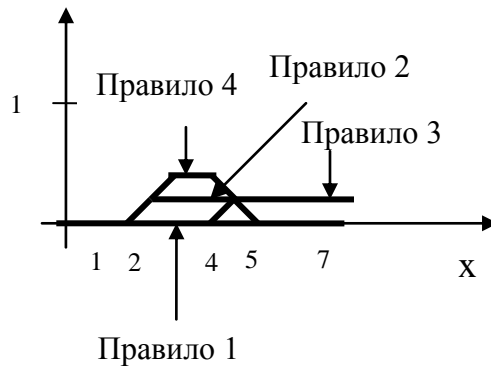
Правило 3: $\min(1, 0.125) = 0.125$;

Правило 4: $\min(0.5, 0.5) = 0.5$.

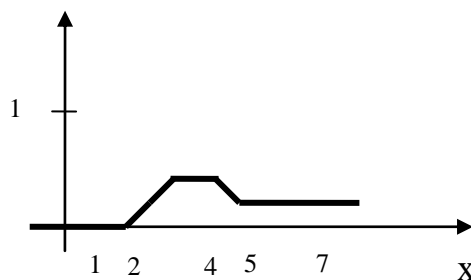
Побудуємо нову вихідну нечітку змінну, використовуючи отримані ступені впевненості:



4. Акумуляція:



Новий терм вихідної змінної. Кількість годин:



5. Виходячи з отриманого графіка ступені приналежності вихідного терма, можна сказати, що Іванову А.А., який має середню оцінку 3.5, на розв'язання 9 варіантів завдань знадобиться не менше ніж 2.75 години (ступінь впевненості даного твердження 0,5).

Задачі для самостійного розв'язання

1. Побудувати нечітку базу знань (використовувати не менше 3 лінгвістичних змінних) для задачі закупівель (співвідношення ціни, якості, обсягу закупівель і т. ін.), перевірити її на повноту і виконати нечіткий вивід для конкретних значень (вибрати випадковим чином).
2. Побудувати нечітку базу знань (використовувати не менше 3 лінгвістичних змінних) для задачі розподілу навантажень спортсмена (співвідношення навантажень, фізичного стану, споживаних калорій і т. ін.), перевірити її на повноту і виконати нечіткий вивід для конкретних значень (вибрати випадковим чином).
3. Побудувати нечітку базу знань (використовувати не менше 3 лінгвістичних змінних) для задачі керування транспортним засобом (регулювання швидкості з урахуванням передачі, погодних умов, інтенсивності потоку і т. ін.), перевірити її на повноту і виконати нечіткий вивід для конкретних значень (вибрати випадковим чином).
4. Побудувати нечітку базу знань (використовувати не менше 3 лінгвістичних змінних) для задачі керування транспортним засобом (управління кермом, газом, гальмом при в'їзді в гараж), перевірити її на повноту і виконати нечіткий вивід для конкретних значень (вибрати випадковим чином).
5. Побудувати нечітку базу знань (використовувати не менше 3 лінгвістичних змінних) для завдання регулювання теплопостачання (співвідношення середньодобової температури, вітру, розміру будівлі тощо), перевірити її на повноту і виконати нечіткий вивід для конкретних значень (вибрати випадковим чином).

6. Побудувати нечітку базу знань (використовувати не менше 3 лінгвістичних змінних) для завдання регулювання реверсного руху на дніпровському мосту (ураховувати час, інтенсивність потоку, день тижня і т. ін.), перевірити її на повноту і виконати нечіткий вивід для конкретних значень (вибрати випадковим чином).
7. Побудувати нечітку базу знань (використовувати не менше 3 лінгвістичних змінних) для задачі підбору спецій для блюда (співвідношення кількості та гостроти спецій, рецептури, переваг їдця, обсягу їжі і т. ін.), перевірити її на повноту і виконати нечіткий вивід для конкретних значень (вибрати випадковим чином).
8. Побудувати нечітку базу знань (використовувати не менше 3 лінгвістичних змінних) для задачі підбору об'єму страв (ураховувати калорійність, смакові переваги, кількість їдців і т.ін.), перевірити її на повноту і виконати нечіткий вивід для конкретних значень (вибрати випадковим чином).
9. Побудувати нечітку базу знань (використовувати не менше 3 лінгвістичних змінних) для задачі подачі електроенергії в умовах економії (облік часу доби, типу приміщень, кількості людей, типу обладнання і т. ін.), перевірити її на повноту і виконати нечіткий вивід для конкретних значень (вибрати випадковим чином).
10. Побудувати нечітку базу знань (використовувати не менше 3 лінгвістичних змінних) для задачі підбору інтенсивності занять (ураховувати початковий рівень підготовки, обсяг навчального матеріалу, кількість осіб у групі, необхідний рівень засвоєння і т.ін.), перевірити її на повноту і виконати нечіткий вивід для конкретних значень (вибрати випадковим чином).
11. Побудувати нечітку базу знань (використовувати не менше 3 лінгвістичних змінних) для задачі розрахунку споживання бензину (ураховувати тип маневрів, рівень підготовки водія, стан автомобіля, тип автомобіля і т. ін.), перевірити її на повноту і

виконати нечіткий вивід для конкретних значень (вибрати випадковим чином).

12. Побудувати нечітку базу знань (використовувати не менше 3 лінгвістичних змінних) для завдання регулювання системи зрошення (ураховувати пору року, кількість опадів, вид зрошуваної культури тощо), перевірити її на повноту і виконати нечіткий вивід для конкретних значень (вибрати випадковим чином).

13. Побудувати нечітку базу знань (використовувати не менше 3 лінгвістичних змінних) для завдання налаштування аудіосистеми (потужність колонок, їх кількість, розмір приміщення, призначення установки і т. ін.), перевірити її на повноту і виконати нечіткий вивід для конкретних значень (вибрати випадковим чином).

14. Побудувати нечітку базу знань (використовувати не менше 3 лінгвістичних змінних) для задачі вибору дози снодійного (кількість препарату, дія препарату, сприйнятливність до вибраного препарату, мета і т. ін.), перевірити її на повноту і виконати нечіткий вивід для конкретних значень (вибрати випадковим чином).

15. Побудувати нечітку базу знань (використовувати не менше 3 лінгвістичних змінних) для завдання планування обсягу виробництва продукції (з урахуванням можливого прибутку, необхідних ресурсів, платоспроможності населення, ринку збуту і т.ін.), перевірити її на повноту і виконати нечіткий вивід для конкретних значень (вибрати випадковим чином).

16. Побудувати нечітку базу знань (використовувати не менше 3 лінгвістичних змінних) для завдання регулювання кондиціонера (ураховувати його потужність, об'єм приміщення, температуру навколишнього середовища, необхідну температуру в приміщенні і т. ін.), перевірити її на повноту і виконати нечіткий вивід для конкретних значень (вибрати випадковим чином).

17. Побудувати нечітку базу знань (використовувати не менше 3 лінгвістичних змінних) для задачі розподілу навантаження між

комп'ютерами при використанні їх у кластерах (ураховувати характеристики комп'ютерів, їх кількість, кількість паралельного коду, характеристики мережі і т. ін.), перевірити її на повноту і виконати нечіткий вивід для конкретних значень (вибрати випадковим чином).

18. Побудувати нечітку базу знань (використовувати не менше 3 лінгвістичних змінних) для задачі вибору складського приміщення (ураховувати площу складу, кількість і розміри продукції, віддаленість від місця виробництва і точок реалізації, якість продукції, характеристики приміщень і т. ін.), перевірити її на повноту і виконати нечіткий вивід для конкретних значень (вибрати випадковим чином).

19. Побудувати нечітку базу знань (використовувати не менше 3 лінгвістичних змінних) для задачі вибору комплектуючих для комп'ютера (ураховувати ціну, потреби користувача, сумісність, терміни використання і т. ін.), перевірити її на повноту і виконати нечіткий вивід для конкретних значень (вибрати випадковим чином).

20. Побудувати нечітку базу знань (використовувати не менше 3 лінгвістичних змінних) для задачі визначення кількості ліній у службі підтримки (ураховувати кількість клієнтів, що обслуговуються, середню частоту звернення до служби одного клієнта, середній час обслуговування однієї заявки, кваліфікацію персоналу і т. ін.), перевірити її на повноту і виконати нечіткий вивід для конкретних значень (вибрати випадковим чином).

Приклад 3.17. Нечіткий регулятор

Розглянемо приклад використання апарату нечіткої логіки в задачі управління замкнутої системи регулювання (u , y , e , x – відповідно, що надходить на вхід регулятора, і вихідний сигнал регулятора):

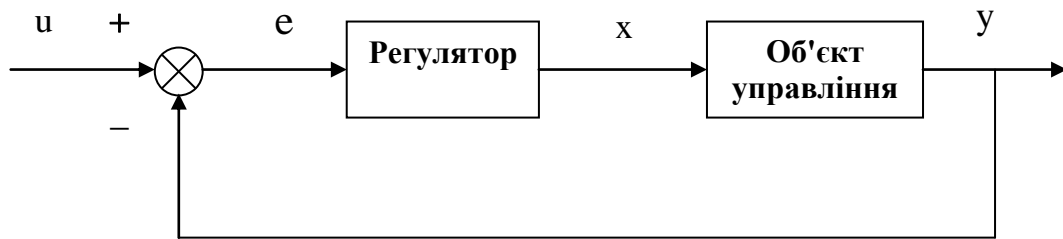


Рис. 3.49. Замкнута система регулювання

У розглянутій системі регулятор виробляє керуючий сигнал x відповідно до обраного алгоритму регулювання, наприклад, пропорційно сигналу помилки, або її інтегралу і т. ін. Покажемо, що для вироблення такого сигналу застосовні методи апарату нечіткої логіки.

Припустимо, що функції регулятора виконує мікроконтролер, при цьому аналоговий сигнал e обмежений діапазоном $[-1,1]$ і перетворює в цифрову форму аналогово-цифровим перетворювачем з дискретністю $0,25$, а вихідний сигнал регулятора x формується з допомогою цифро-аналогового перетворювача і має всього 5 рівнів: $-1, -0,5, 0, 0,5, 1$.

Беручи до уваги дані рівні, уведемо лінгвістичні змінні:

A_1 : великий позитивний,

A_2 : малий позитивний,

A_3 : нульовий,

A_4 : малий негативний,

A_5 : великий негативний,

і на дискретній множині можливих значень сигналу неузгодженості e визначимо функції належності так, як це наведено в табл.3.1.

Припустимо, далі, що функціонування регулятора визначається такими правилами:

П1: якщо $e = A_3$ і $\Delta e = A_3$, то $x = 0$,

П2: якщо $e = A_2$ і $\Delta e = A_2$, то $x = 0,5$,

П3: якщо $e = A_4$, і $\Delta e = A_4$, то $x = 1$,

П4: якщо $e = A_1$, і $\Delta e = A_1$, то $x = -1$,

де Δe – перша різниця сигналу помилки в поточний дискретний момент часу.

Таблиця 3.1. Значення функції приналежності

	-1	-0,75	-0,5	-0,25	0	0,25	0,5	0,75	1
$A_1(e)$	0	0	0	0	0	0	0,3	0,7	1
$A_2(e)$	0	0	0	0	0,3	0,7	1	0,7	0,3
$A_3(e)$	0	0	0,3	0,7	1	0,7	0,3	0	0
$A_4(e)$	0,3	0,7	1	0,7	0,3	0	0	0	0
$A_5(e)$	1	0,7	0,3	0	0	0	0	0	0

Відзначимо, що набір правил може бути, і якимось іншим. Якщо, наприклад, використовується спрощений алгоритм нечіткого виведення, то призначеннях, скажімо $e = 0,25$ і $\Delta e = 0,5$ маємо:

$$\alpha_1 = \min(0,7; 0,3) = 0,3u \quad x_1 = 0,$$

$$\alpha_2 = \min(0,7; 1) = 0,7u \quad x_2 = -0,5,$$

$$\alpha_3 = \min(0; 0) = 0u \quad x_3 = 1,$$

$$\alpha_4 = \min(0; 0,3) = 0,3u \quad x_4 = -1,$$

і вихід регулятора

$$x = \frac{0,3 \cdot 0 + 0,7 \cdot (-0,5) + 0 \cdot 1 + 0,3 \cdot (-1)}{0,3 + 0,7 + 0 + 0,3} = \frac{-0,65}{1,3} = -0,5.$$

Аналогічним чином значення вихідного сигналу регулятора розраховуються при інших значеннях e і Δe .

Зазначимо, що при проектуванні подібних регуляторів основним етапом є завдання набору нечітких правил. Інші аспекти: вибір форми функцій приналежності, алгоритму приведення до чіткості.

3.8. Побудова нечітких моделей в середовищі MATLAB Fuzzy Logic Toolbox

3.8.1. Призначення та можливості пакета Fuzzy Logic Toolbox

Пакет Fuzzy Logic Toolbox (пакет нечіткої логіки) – це сукупність прикладних програм, що відносяться до теорії розмитих або нечітких множин і дозволяють конструювати так звані нечіткі експертні та/або керуючі системи.

Основні можливості пакету:

- побудова систем нечіткого виводу (експертних систем, регуляторів, апроксиматорів залежностей);
- побудова адаптивних нечітких систем (гібридних нейронних мереж);
- інтерактивне динамічне моделювання в Simulink.

Пакет дає можливість працювати:

- у режимі графічного інтерфейсу;
- у режимі командного рядка;
- з використанням блоків і прикладів пакету Simulink.

3.8.2. Графічний інтерфейс FuzzyLogicToolbox

Склад графічного інтерфейсу. До складу програмних засобів Fuzzy Logic Toolbox входять такі основні програми, що дозволяють працювати в режимі графічного інтерфейсу: редактор нечіткої системи виведення Fuzzy Inference System Editor (FIS Editor або FIS-редактор) разом з допоміжними програмами – редактором функцій приналежності (Membership Function Editor), редактором правил (Rule Editor), переглядачем правил (Rule Viewer) і переглядачем поверхні відгуку (SurfaceViewer); редактор гібридних систем (ANFIS Editor, ANFIS-редактор).

Для розробки та подальшого застосування систем нечіткого виводу можуть бути використані такі графічні засоби, що входять до складу пакета Fuzzy Logic Toolbox:

- редактор систем нечіткого виводу FIS;
- редактор функцій приналежності систем нечіткого виводу;
- редактор правил систем нечіткого виводу;
- редактор перегляду правил систем нечіткого виводу;
- програма перегляду поверхні системи нечіткого виводу;

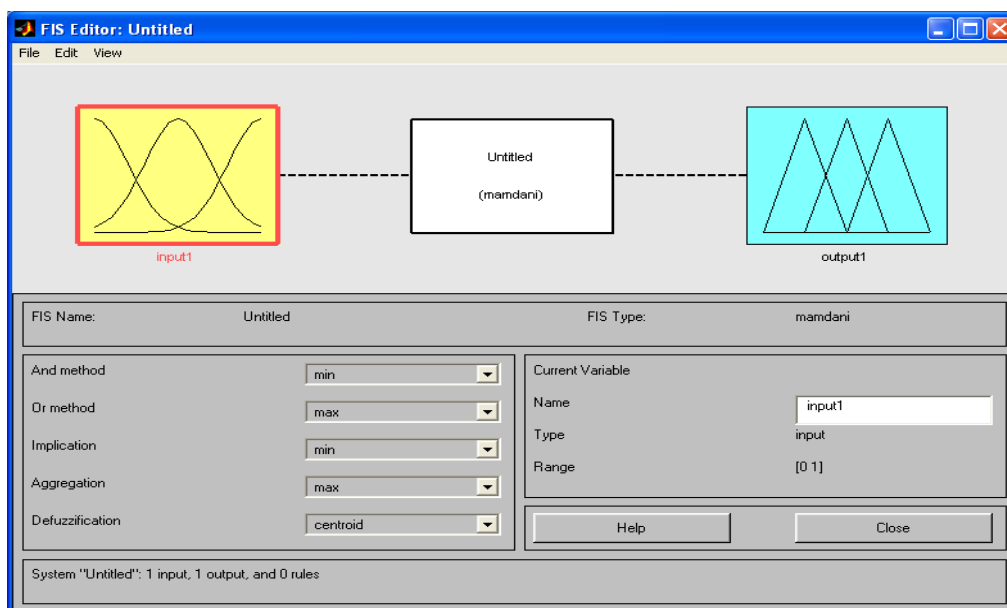


Рис.3.50. Графічний інтерфейс редактора FIS, що викликається функцією fuzzy

Приклад 3.18. Побудова нечіткої апроксимуючої системи. Командою (функцією) Fuzzy з режиму командного рядка запускається основна інтерфейсна програма пакету Fuzzy Logic–редактор нечіткої системи виводу (Fuzzy Inference System Editor, Fis Editor, Fis-редактор). Вигляд вікна при цьому показаний на рис. 3.50.

Головне меню редактора містить позиції: **File** – робота з файлами моделей (їх створення, збереження, читання і друк); **Edit** – редагування (додавання та вилучення вхідних і вихідних змінних); **View** – перехід до додаткового інструментарію.

Сконструємо нечітку систему, що відображає залежність між змінними x і y , задану таблицею.

Таблиця 3.2. Значення x і y .

X	1	0,6	0,0	0,4	1
Y	-1	0,36	0,0	0,16	1

Представлені в таблиці дані відображають залежність $y = x^2$. Необхідні дії відобразимо такими пунктами.

1. У позиції меню File вибираємо опцію NewSugenoFIS (нова система типу Sugeno), при цьому в блоці, який відображається білим квадратом, у верхній частині вікна редактора з'явиться напис Untitled2 (Sugeno).

2. Клацнемо лівою кнопкою миші по блоку, що має назву input1 (входи). Потім у правій частині редактора в полі, названому Name (Ім'я), замість input1 введемо позначення нашого аргументу, тобто x . Звернемо увагу, що якщо тепер зробити де-небудь (поза блоків редактора) одноразове клацання миші, то ім'я зазначеного блоку зміниться на ж; те ж досягається натисненням після введення клавіші Enter.

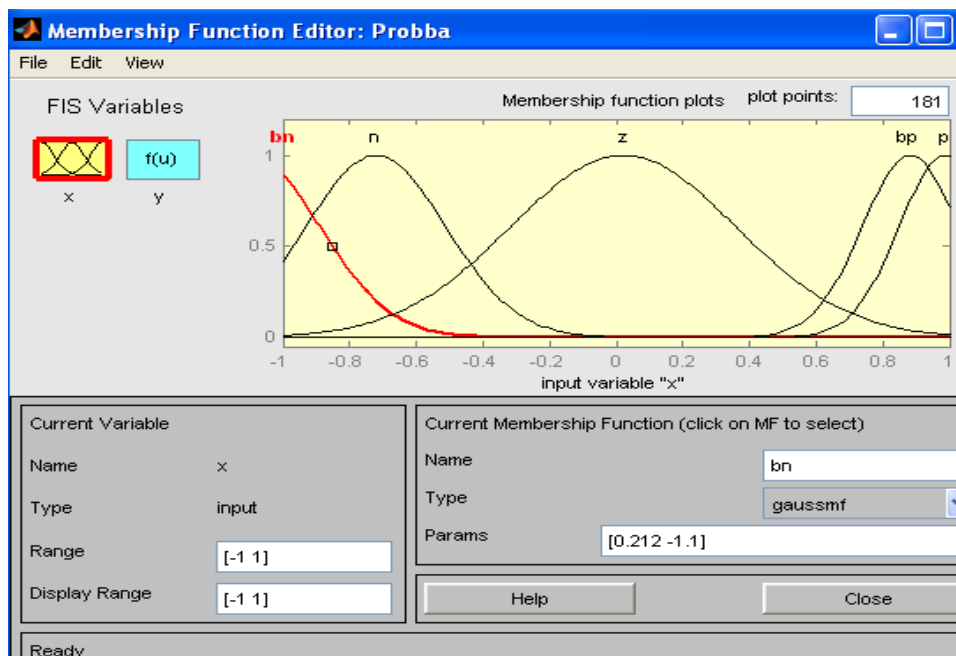


Рис.3.51. Вікно редактора функцій належності

3. Двічі клацнемо по цьому блоку. Перед нами відкриється вікно редактора функцій приналежності – Membership Function Editor

(рис. 3.51). Увійдемо в позицію меню Edit даного редактора і виберемо в ньому опцію Add MFs (Add Membership Functions— додати функції приналежності). При цьому з'явиться діалогове вікно (рис.3.52), що дозволяє задати тип (MFtype) і кількість (Number of MFs) функцій приналежності (в даному випадку все відноситься до вхідного сигналу, тобто до змінної x). Виберемо гауссові функції приналежності (gaussmf), а їх кількість задамо рівним п'яти — по числу значень аргументу в табл. Підтвердимо введення інформації натисканням кнопки ОК, після чого відбудеться повернення до вікна редактора функцій приналежності.

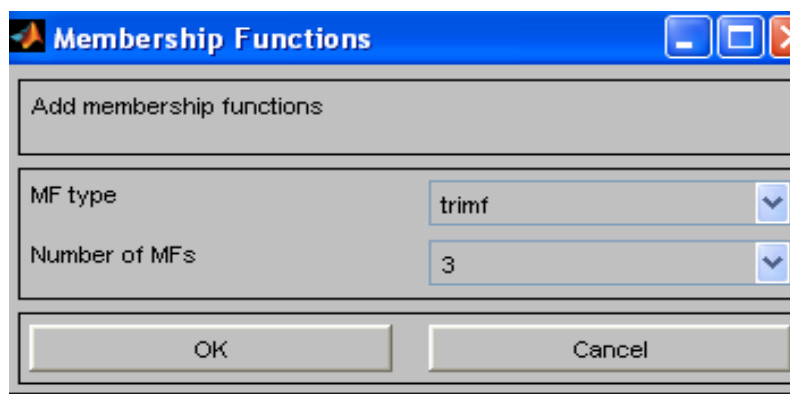


Рис.3.52. Діалогове вікно завдання типу і кількості функцій належності

4. У полі Range (Діапазон) встановимо діапазон зміни x від -1 до 1 , тобто діапазон, відповідний табл. Потім клацнемо лівою кнопкою миші де-небудь у полі редактора (або натиснемо клавішу вводу Enter). Звернемо увагу, що після цього станеться відповідна зміна діапазону у полі Display Range (Діапазон дисплея).

5. Звернемося до графіків заданих нами функцій приналежності, зображеним у верхній частині вікна редактора функцій приналежності. Зауважимо, що для успішного вирішення поставленої задачі необхідно, щоб ординати максимумів цих функцій збігалися з заданими значеннями аргументу x . Для лівої, центральної і правої функцій таку умову виконано, але дві інші необхідно «посунути» вздовж осі абсцис. Підводимо курсор до

потрібної кривої і клацаємо лівою кнопкою миші. Крива вибирається, зафарбовуючись у червоний колір, після чого за допомогою курсору її можна посунути в потрібну сторону (більш точну установку можна провести, змінюючи числові значення в полі Params – у даному випадку кожній функції приналежності відповідають два параметри, при цьому перший визначає розмах кривої, а другий – положення її центру). Для обраної кривої в полі Name можна змінити ім'я. Зробимо необхідні переміщення кривих і поставимо всім п'яти кривим нові імена, наприклад:

- самій лівій – bn,
- такій – n,
- центральній – z,
- такої за нею праворуч – p,
- самій лівій – bp.

Натиснути кнопку Close.

6. Зробити одноразове клацання по блакитному блоку (output1). У вікні Name замінимо ім'я output1 на y.

7. Двічі клацнемо по зазначеному блоку і перейдемо до редактора функцій приналежності. У позиції меню Edit виберемо опцію Fdd MFs. З'являється вікно (рис.3.52), що дозволяє задати в якості функції приналежності тільки лінійні або постійні залежно від того, який алгоритм Sugeno вибирають. У даній задачі необхідно вибрати постійні функції приналежності з загальним числом 4. Введені дані підтверджують натисканням кнопки ОК.

8. Змінимо імена функцій належності (їх графіки при використанні алгоритму Sugeno для вихідних змінних не наводяться), наприклад, задавши їх як відповідні числові значення y, тобто 0, 0,16, 0,36, 1; одночасно ці ж числові значення введемо в поле Params (рис.3.53). Закриємо вікно кнопкою Close.

9. Двічі клацнемо по середньому блоку – відкривається редактор правил (RuleEditor). Введемо відповідні правила. При введенні кожного правила необхідно визначити відповідність між кожною функцією приналежності аргументу x і числовим значенням y. Крива, позначена нами bn, відповідає $x = -1$, тобто $y = 1$. Виберемо, потім у лівому полі з заголовком x is) bn, а в правому

1 і натиснемо кнопку Addrule (додати правило). Уведене правило з'явиться у вікні правил і буде представляти собою запис: 1.If (x isbn) then (y is 1) (1). Аналогічно поступимо для всіх інших значень x, у результаті чого сформується набір з 5 правил (див. рис. 3.54). Закриємо вікно редактора правил і повернемося у вікно FIS-редактора. Побудову системи закінчено і можна почати експерименти по її дослідженню. Зауважимо, що більшість опцій обрана нами за замовчуванням.

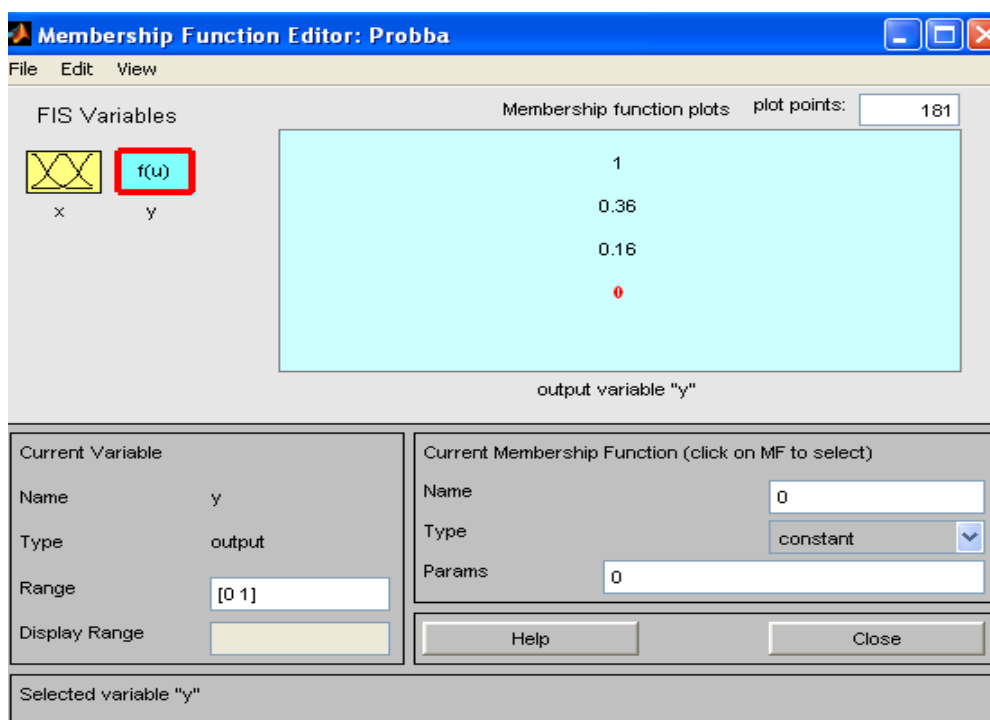


Рис.3.53. Параметри функцій належності змінної у

10. Попередньо збережемо на диску (використовуючи пункти меню File/Savetodiskas...) створену систему під яким-небудь ім'ям, наприклад, Proba.

11. Виберемо позицію меню View. Як видно з підменю, що випадає при цьому, за допомогою пунктів Edit membership functions і Editrules можна здійснити перехід до двох вище розглянутих програм – редакторів функцій приналежності та правил (те ж можна зробити і натисканням клавіш Ctrl+2 або Ctrl+3), але зараз нас будуть цікавити два інших пункти – Viewrules (перегляд правил) і Viewsurface (Перегляд поверхні).

Виберемо пункт Viewrules, при цьому відкриється вікно (рис. 3.55) ще однієї програми – перегляду правил (RuleViewer).

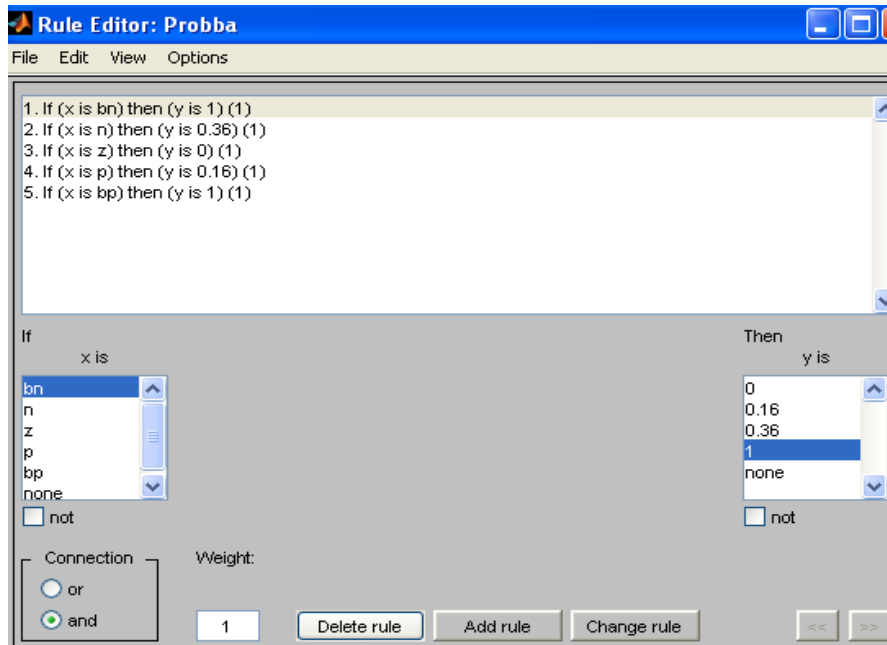


Рис.3.54. Вікно редактора правил

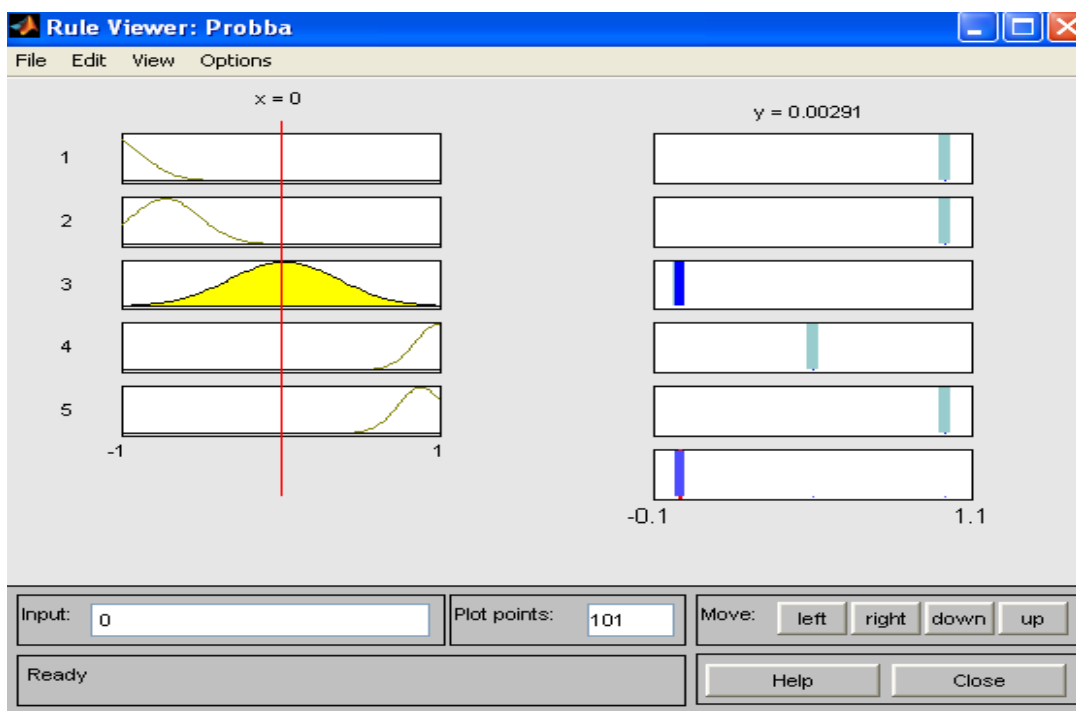


Рис.3.55. Вікно перегляду правил

12. У правій частині вікна в графічній формі представлені функції приналежності аргументу x , у лівій – змінної виходу y з поясненням механізму прийняття рішення. Червона вертикальна лінія, що перетинає графіки у правій частині вікна, яку можна переміщати за допомогою курсора, дозволяє змінювати значення змінної входу (це можна робити, задаючи числові значення в полі Input (Вхід), при цьому відповідно змінюються значення у правій верхній частині вікна. Задамо, наприклад, $x = 0.5$ в полі Input і натиснемо клавішу вводу (Enter). Значення відразу зміниться і стане рівним 0.202. Таким чином, за допомогою побудованої моделі і вікна перегляду правил можна вирішувати задачу інтерполяції, тобто задачу, рішення якої і було потрібно знайти. Зміна аргументу шляхом переміщення червоної вертикальної лінії дуже наочно демонструє, як система визначає значення виходу.

13. Закриємо вікно перегляду правил і вибором пункту меню View/View surface перейдемо до вікна перегляду поверхні відгуку (виходу), у нашому випадку – до перегляду кривої (рис.3.56).

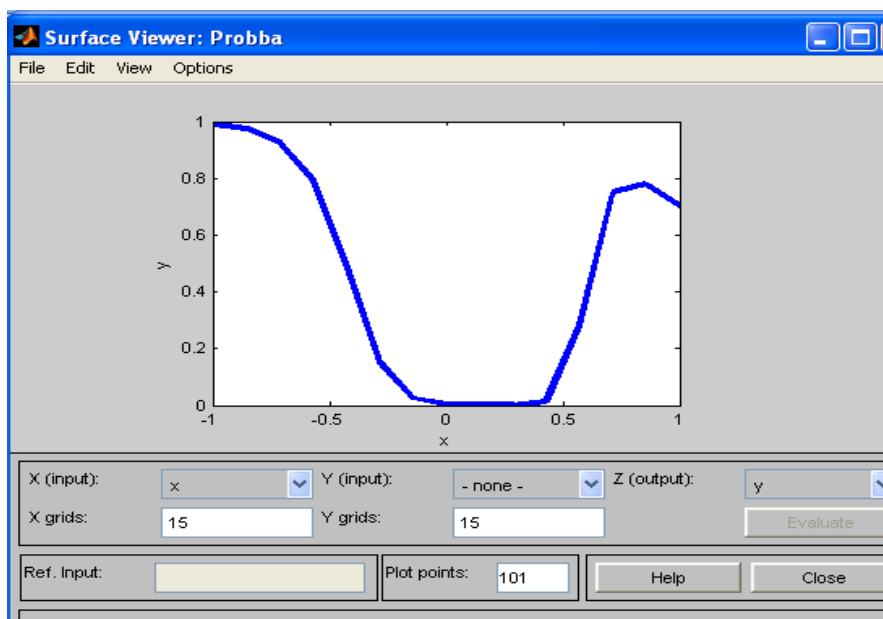


Рис.3.56. Вікно перегляду поверхні відгуку

Видно, що змодельоване системою за таблицею даних відображення не дуже нагадує функцію. Ну що ж, нічого дивного в цьому немає: число експериментальних точок невелике, а параметри функцій приналежності (x) обрані, швидше за все, неоптимальним чином. Нижче ми розглянемо можливість поліпшення якості подібної моделі за допомогою гібридних (нечітких) нейронних систем (редактор гібридних систем, див. 3.10).

На закінчення розгляду прикладу зазначимо, що з допомогою вищевказаних програм-редакторів на будь-якому етапі проектування нечіткої моделі до неї можна внести необхідні корективи, аж до завдання якої-небудь особливої користувацької функції приналежності. З опцій, що встановлюються у FIS-редакторі за замовчуванням при використанні алгоритму Sugeno, можна відзначити:

- логічний вивід організується за допомогою операції множення (`prod`);
- композиція – за допомогою операції логічної суми (імовірнісного АБО, `probog`);
- приведення до чіткості – дискретним варіантом центроїдного методу (зваженим середнім, `wtaver`).

Використовуючи відповідні поля у лівій нижній частині вікна FIS-редактора, дані опції можна, за бажання, змінити. Таким чином, нечітке моделювання в середовищі MATLAB здійснюється з використанням пакета розширення Fuzzy Logic Toolbox, в якому реалізовані десятки функцій нечіткої логіки та нечіткого виводу. Користувач може виконувати необхідні дії по розробці і використанню нечітких моделей в одному з таких режимів:

- в ітеративному режимі з допомогою графічних засобів редагування і візуалізації всіх компонентів систем нечіткого виводу;
- у режимі команд за допомогою введення імен відповідних функцій з необхідними аргументами безпосередньо у вікно команд системи MATLAB.

Редактор систем нечіткого виводу FIS

Редактор систем нечіткого виводу FIS є основним засобом, який використовується для створення або редагування систем нечіткого виводу у графічному режимі. Редактор FIS може бути відкритий за допомогою введення функції `fuzzy` у вікні команд. Ця функція надає користувачу можливість задавати і редагувати на високому рівні властивості системи нечіткого виводу, такі як число вхідних і вихідних змінних, тип системи нечіткого виведення, використовуваний метод дефазифікації і т. ін.

Якщо функція `fuzzy` викликається без аргументів, то редактор FIS викликається для новостворюваної системи нечіткого виводу з ім'ям `Untitled` за замовчуванням (рис.3.46). При цьому за замовчуванням використовується також ряд параметрів, таких як тип системи виводу (Мамдані), нечіткі логічні операції, методи імплікації, агрегування та дефазифікації та й деякі інші. Користувач може погодитися з цими значеннями або змінити їх. Редактор FIS має графічний інтерфейс і дозволяє викликати всі інші редактори і програми систем нечіткого виводу.

У верхній частині робочого інтерфейсу редактора зображується діаграма, яка представляє у візуальній формі входи і виходи системи нечіткого виведення, у центрі яких знаходиться так званий процесор нечітких правил. Клацання на прямокутнику із зображенням входу або виходу виділяє відповідну змінну і робить її поточною (виділяється червоним кольором). Подвійне клацання на прямокутнику із зображенням вхідної або вихідної змінної викликає редактор функцій приналежності із завантаженою у нього відповідною змінною. Подвійне клацання на зображенні процесора нечітких правил викликає редактор правил для відповідної системи нечіткого виводу. Якщо деяка змінна існує в системі нечіткого виводу, але не використовується правила виводу, то зв'язок її з процесором нечітких правил зображується не суцільний, а пунктирною лінією.

Пункт меню `File` редактора FIS містить такі операції:

- **NewFis** Дозволяє вибрати тип системи нечіткого виводу: Mamdani або Sugeno. При цьому задана система нечіткого виводу не має ні вхідних, ні вихідних змінних.

- **Import** – дозволяє завантажити в редактор FIS існуючу систему нечіткого виводу одним з таких способів: FromWorkspace... – з робочого простору програми MATLAB або FromDisk... - з зовнішнього файлу.

- **Export** – дозволяє зберегти редаговану систему нечіткого виводу: ToWorkspace... – у робочому просторі програми MATLAB або ToDisk... – у зовнішньому файлі.

- **Close** – закриває редактор.

Пункт меню Edit містить такі операції:

- **Undo** – скасовує виконання останньої дії;

- **Add Variable....** – дозволяє додати до редагованої системи нечіткого виводу змінну одного з таких типів: Input - вхідну змінну або Output – вихідну змінну;

- **Remove Selected Variable** – видалити вибрану змінну;

- **Membership Function...** – викликає редактор функцій приналежності;

- **Rules** – викликає редактор правил нечіткого виводу;

Пункт меню View містить такі операції:

- **Rules** – викликає програму перегляду правил нечіткого виводу;

- **Surface** – викликає програму перегляду поверхні нечіткого виводу.

У лівій нижній частині робочого інтерфейсу редактора є 5 спливаючих меню:

And method (Метод логічної кон'юнкції) – дозволяє задати один з таких методів для виконання логічної кон'юнкції в умовах нечітких правил:

- min – метод мінімального значення;
- prod – метод алгебраїчного добутку;
- Custom – метод, визначений користувачем.

Ormethod (Метод логічної диз'юнкції) – дозволяє задати один з таких методів для виконання логічної диз'юнкції в умовах нечітких правил:

- max – метод максимального значення;
- probor – метод алгебраїчної суми;
- Gustom - метод, визначений користувачем;

Implication method (Метод виведення висновку) – дозволяє задати один з таких методів для виконання (активізації) логічного висновку в кожному з нечітких правил:

- min – метод мінімального значення;
- prod – метод алгебраїчного добутку;
- Gustom - метод, визначений користувачем.

Aggregation method (Метод агрегування) – дозволяє задати один з таких методів агрегування значень функції приналежності кожної з вихідних змінних у виводах нечітких правил:

- max – метод максимального значення;
- sum – метод граничної суми;
- probor – метод алгебраїчної суми;
- Gustom - метод, визначений користувачем.

Defuzzification method (Метод дефазифікації) – дозволяє задати один з таких методів для виконання дефазифікації вихідних змінних у системі нечіткого виводу типу Мамдані;

- centroid – метод центру тяжіння для дискретної множини значень функції приналежності;
- bisektor – метод центру площі;
- pom – метод середнього максимуму, яка визначається як середнє арифметичне лівого і правого модальних значень;
- som – метод найменшого (лівого) модального значення;
- lom – метод найбільшого (правого) модального значення;
- Gustom – метод, визначений користувачем.

Для систем нечіткого виводу типу Сугено можна обрати один з таких методів дефазифікації:

- wtaver – метод зваженого середнього;
- wtsum – метод зваженої суми.

Редактор функції належності

Редактор функцій приналежності призначений для завдання і редагування функцій приналежності окремих термів системи нечіткого виводу у графічному режимі. Редактор може бути відкритий введенням функції `mfedit` або натисканням клавіш `<Ctrl>+<2>`.

Для кожної функції приналежності можна змінювати її ім'я, тип і параметри. Користувач має можливість вибрати будь-яку з 11 вбудованих функцій приналежності, а також задати власну. Результат виклику редактора функцій належності за допомогою функції `mfedit` ('tipper') показаний на рис.3.57.

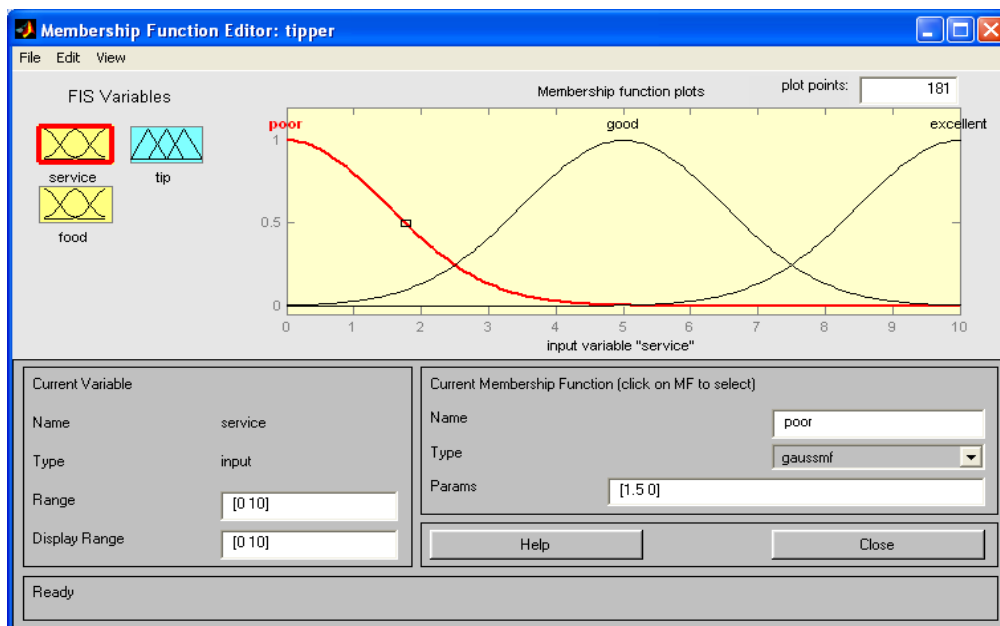


Рис.3.57. Редактор функцій приналежності, що викликається функцією `mfedit` ('tipper')

Для відображення графіків функцій приналежності слід вибрати необхідну змінну у лівій частині графічного інтерфейсу редактора під заголовком **Fis Variables**. Щоб вибрати потрібну функцію приналежності, слід клацнути на ній або її мітці в основному вікні з графіками функцій приналежності.

Пункт меню File редактор функцій приналежності містить такі операції, що і відповідний пункт редактора Fis.

Пункт меню Edit містить такі операції:

- **Undo** – скасовує виконання останньої дії;
- **AddMF ...** – дозволяє додати вбудовану функцію належності термів для обраної змінної;
- **AddCustomMF...** – дозволяє додати власну функцію належності термів для окремої змінної;
- **RemoveCurrentMF** – дозволяє видалити окрему функцію приналежності;
- **RemoveAllMFs** – дозволяє видалити всі функції приналежності для окремої змінної;
- **FisProperties** – викликає редактор Fis.

Пункт меню View містить такі операції:

- **Rules** – викликає програму перегляду правил нечіткого виводу;
- **Surface** – викликає програму перегляду поверхні нечіткого виводу.

Розкритий список типів функцій належності дозволяє вибрати одну з 11 вбудованих функцій приналежності. Використовуючи відповідні поля введення, можна змінити імена термів обраної змінної в поле введення **Name**, модифікувати параметри вбудованих функцій приналежності в полі вводу **Params**.

Редактор правил системи нечіткого виводу

Редактор правил системи нечіткого виводу призначений для завдання і редагування окремих правил системи нечіткого виводу у графічному режимі. Він може бути відкритий введенням функції ruleedit ('a') або натисканням клавіш <Ctrl>+<3>. Ця функція дозволяє також виконувати граматичний аналіз правил.

Щоб використовувати даний редактор для створення правил, необхідно попередньо визначити всі вхідні і вихідні

змінні. Задати правила можна за допомогою вибору відповідних значень термів вхідних і вихідних змінних.

Результат виклику функції ruleedit ('tipper') зображено на рис. 3.58.

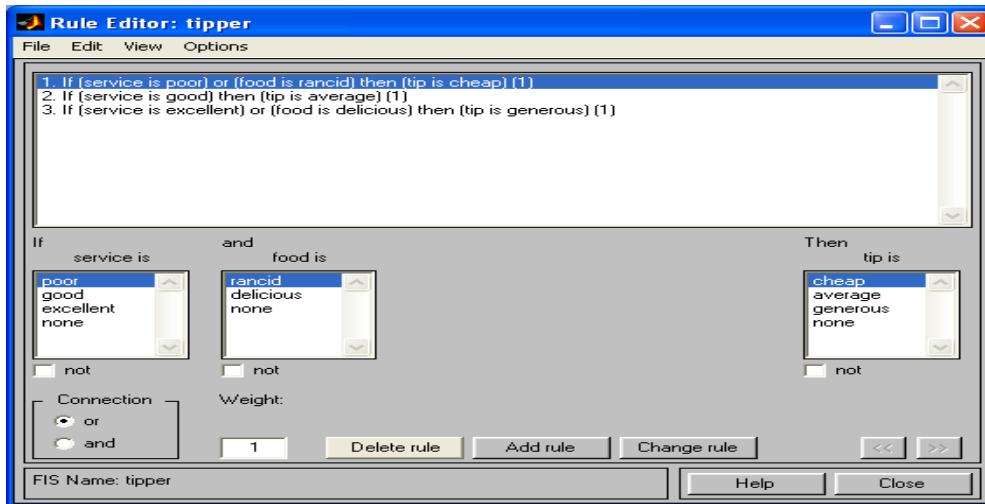


Рис.3.58. Редактор правил, що викликається функцією ruleedit ('tipper')

Пункт меню File редактор функцій приналежності містить такі операції, що і відповідний пункт редактора Fis. Пункт меню Edit містить такі операції:

- **Undo** – скасовує виконання останньої дії;
- **Fis Properties** – викликає редактор Fis;
- **Membership Functions...** – викликає редактор функцій приналежності.

Пункт меню View містить такі операції:

- **Rules** – викликає програму перегляду правил нечіткого виводу;
- **Surface** – викликає програму перегляду поверхні нечіткого виводу.

Пункт меню Options містить такі операції:

- **Language** – дозволяє вибрати мову для запису правил у формі тексту;

• **Format** – дозволяє вибрати формат запису правил системи нечіткого виводу: Verbose (у формі тексту), Symbolic (у символічній формі) або Indexed (у цифровій формі). При запису правил у формі тексту для створення закінчених пропозицій використовуються службові слова "if", "then", "is", "AND", "OR" і т.д. При запису правил у символічній формі ці службові слова замінюються символами відповідних операцій. Поля вводу в середній частині графічного інтерфейсу редактора правил дозволяють задати нове правило в системі нечіткого виводу. Для цього необхідно виділити ім'я терма відповідної змінної, яка повинна бути попередньо визначена за допомогою редактора функцій приналежності. Якщо в умові правила використовується логічне заперечення деякого терма, то для цього терма слід зазначити відповідний прапорець з міткою "not". Редактор правил дозволяє встановити логічні зв'язки для підумов правила (**Connection**) і ваги правила поле вводу **Weight**).

Програми перегляду правил і поверхні системи нечіткого виводу

Графічний інтерфейс програми перегляду правил може бути відкритий за допомогою введення функції ruleview('a'), або за допомогою головного меню редактора Fis, редактора функцій приналежності, або редактора правил, або натисканням клавіш <Ctrl>+<5>.

Програма перегляду правил не дозволяє редагувати правила і функції належності термів змінних і використовується після розробки системи нечіткого виводу на етапі її аналізу та оцінки. Користувач має можливість оцінити вихідні значення змінних нечіткої моделі і вплив кожного з правил на результат нечіткого виводу шляхом зміни значень вхідних змінних. Графічний інтерфейс програми зображено на рис.3.59.

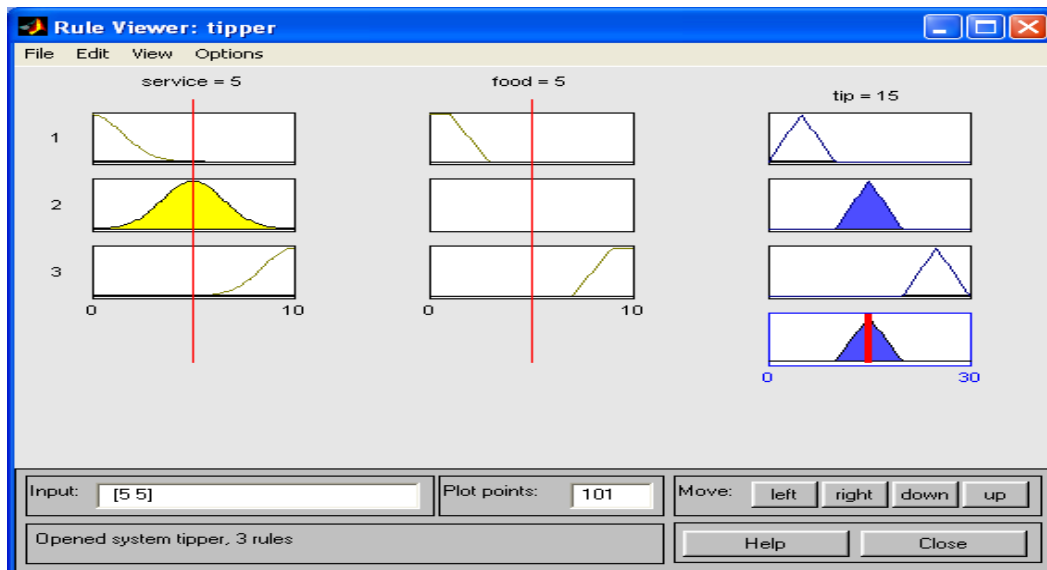


Рис.3.59. Програма перегляду правил, викликана функцією ruleview ('tipper')

Пункт меню **File** редактор функцій приналежності містить такі операції, що і відповідний пункт редактора **Fis**. Пункт меню **Edit** містить такі операції:

- **Undo** – скасовує виконання останньої дії;
- **Fis Properties** – викликає редактор Fis;
- **Membership Functions...** – викликає редактор функцій приналежності;
- **Rules** – викликає програму перегляду правил нечіткого виводу.

Пункт меню **View** містить такі операції:

- **Surface** – викликає програму перегляду поверхні нечіткого виводу.

Пункт меню **Options** містить такі операції:

- **Format** – дозволяє вибрати формат запису правил системи нечіткого виводу: **Verbose** (у формі тексту), **Symbolic** (у символічній формі) або **Indexed** (у цифровій формі).

У центральній частині графічного інтерфейсу програми перегляду правил розташовані прямокутники, відповідні окремим вхідним змінним (функції приналежності жовтого кольору) і

вихідним змінним (функції приналежності синього кольору) правил нечіткого виводу. При цьому кожному правилу відповідає окремий рядок з цих прямокутників. Номери правил вказані у лівій частині графічного інтерфейсу.

У правій нижній частині графічного інтерфейсу розташований прямокутник, який зображає дефазифікацію вихідної змінної після акумулювання всіх виводів правил нечіткого виводу. Отримане в результаті дефазифікації значення вихідної змінної вказується у верхній частині стовпчика з ім'ям цієї вихідної змінної (tip=15 на рис.3.59)

Прямокутники вхідних змінних перетинає вертикальна пряма червоного кольору, положення якої відповідає конкретному значенню вхідної змінної відповідного стовпця. Задати конкретні значення вхідних змінних можна або з допомогою їх запису в полі вводу Input, або з допомогою миші, переміщаючи вертикальні прямі у потрібному напрямку.

Програми перегляду поверхні системи нечіткого виводу дозволяють переглядати поверхню системи нечіткого виводу і візуалізувати графіки залежності вхідних змінних від окремих вхідних змінних. Графічний інтерфейс програми перегляду правил може бути відкритий за допомогою введення функції surfview('a') або за допомогою головного меню редактора Fis, редактора функцій приналежності або редактора правил або натисканням клавіш <Ctrl>+<6>.

Графічний інтерфейс програми перегляду поверхні зображено на рис.3.60.

Пункт меню **File** редактора функцій приналежності містить такі операції, що і відповідний пункт редактора **Fis**. Пункт меню **Edit** містить такі операції:

- **Undo** – скасовує виконання останньої дії;
- **Fis Properties** – викликає редактор Fis;
- **Membership Functions...** – викликає редактор функцій приналежності;

- **Rules** – викликає програму перегляду правил нечіткого виводу.

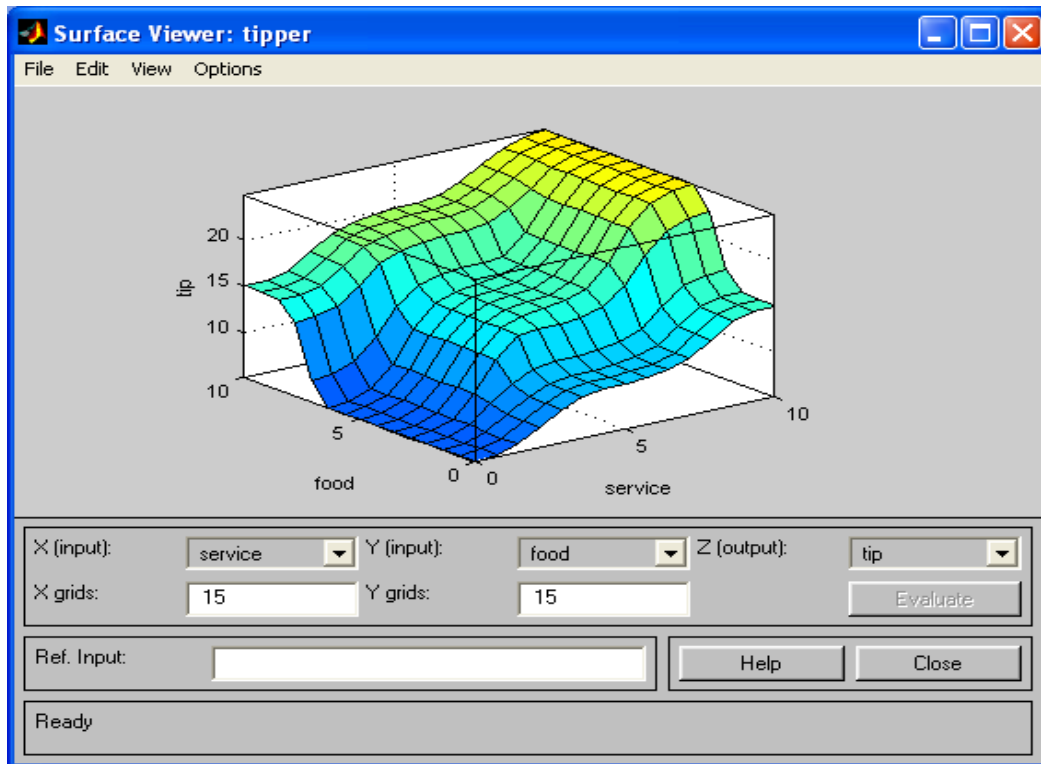


Рис.3.60. Програма перегляду поверхні виводу, що викликана функцією surfview ('tipper')

Пункт меню **View** містить такі операції. Пункт меню **File** редактора функцій приналежності містить такі операції, що і відповідний пункт редактора Fis.

Пункт меню **Edit** містить такі операції:

- **Undo** – скасовує виконання останньої дії;
- **Fis Properties** – викликає редактор Fis;
- **Membership Functions...** – викликає редактор функцій приналежності;
- **Rules** – викликає програму перегляду правил нечіткого виводу.

Пункт меню **View** містить такі операції:

- **Rules** – викликає програму перегляду правил нечіткого виводу.

Пункт меню Options містить такі операції:

- **Plot** – дозволяє вибрати один з 8 стилів зображення графіка поверхні виведення;

- **ColorMap** – дозволяє вибрати одну з 4-колірних схем зображення графіка поверхні виведення.

Програма перегляду поверхні виведення не дозволяє вносити зміни у систему нечіткого виводу і відповідну їй структуру Fis. Використовуючи головне меню програми, користувач може вибрати вхідні змінні і відповідні їм горизонтальні осі системи координат (X і Y), а також вихідну змінну, якій відповідає вертикальна вісь системи координат (Z).

3.9. Нечітке моделювання при рішенні задач управління та прийняття рішень

Одним з основних напрямків практичного використання систем нечіткого виводу є рішення задач управління різними об'єктами або процесами. У цьому випадку побудова нечіткої моделі ґрунтується на формальному представленні характеристик досліджуваної системи у термінах лінгвістичних змінних. Оскільки окрім алгоритму управління, основними поняттями систем управління є вхідні і вихідні змінні, то саме вони розглядаються як лінгвістичні змінні при формуванні бази правил системи нечіткого виводу.

У загальному випадку мета управління полягає в тому, щоб на основі аналізу поточного стану об'єкта керування визначити значення керуючих змінних, реалізація яких дозволяє забезпечити бажану поведінку або стан об'єкта управління. Для вирішення відповідних завдань використовується загальна теорія управління, в рамках якої розроблені різні алгоритми знаходження оптимальних законів управління об'єктами різної

фізичної природи. Базова архітектура або модель класичної теорії управління ґрунтується на уявленні об'єкта і процесу управління у формі деяких систем (рис.3.61).

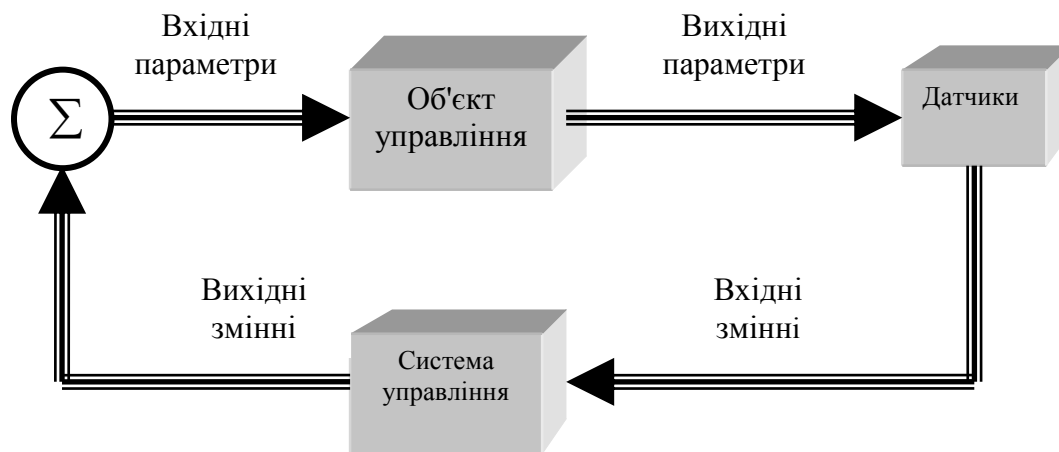


Рис.3.61. Архітектура компонентів процесу управління зі зворотним зв'язком

У загальному випадку мета управління полягає в тому, щоб на основі аналізу поточного стану об'єкта керування визначити значення керуючих змінних, реалізація яких дозволяє забезпечити бажану поведінку або стан об'єкта управління. Для вирішення відповідних завдань використовується загальна теорія управління, в рамках якої розроблені різні алгоритми знаходження оптимальних законів управління об'єктами різної фізичної природи. Базова архітектура або модель класичної теорії управління ґрунтується на уявленні об'єкта і процесу управління у формі деяких систем (рис.3.61).

При цьому об'єкт управління характеризується деякою кінцевою множиною вхідних параметрів і кінцевою множиною вихідних параметрів. На вхід системи управління надходять деякі вхідні змінні, які формуються з допомогою кінцевої множини датчиків. На виході системи управління з використанням деякого алгоритму управління формується множина значень вихідних змінних, які називають керуючими змінними або змінними

процесу управління. Значення цих вихідних змінних надходять на вхід об'єкта управління і, комбінуючись зі значеннями вхідних параметрів об'єкта управління, змінюють його поведінку у бажаному напрямку. Така архітектура називається процесом управління зі зворотним зв'язком, а використовувані для управління технічними об'єктами системи управління – контролерами.

Типовим прикладом розглянутої моделі управління є *пропорційний інтегрально-диференціюючий* контролер або **PID**-контролер. Алгоритм його управління заснований на порівнянні вихідних параметрів об'єкта управління з деякими заданими параметрами і визначенні величини розбіжності між ними або помилки. Після цього розраховуються величини вихідних змінних у формі адитивної суми величини цієї помилки, значення інтегралу і похідної по часу протягом деякого проміжку часу.

Недоліком **PID**-контролерів є допущення лінійного характеру залежності вхідних і вихідних змінних процесу управління, що істотно знижує адекватність цієї моделі при вирішенні окремих практичних завдань. Інший недолік моделі пов'язаний зі складністю виконання відповідних розрахунків, що може призвести до недопустимих затримок у реалізації керуючих впливів при оперативному управлінні об'єктами з високою динамікою зміни вихідних параметрів.

Модель нечіткого управління заснована на заміні класичної системи управління, в якості якої використовуються системи нечіткого виводу. В цьому випадку модель нечіткого управління (рис.3.62) будується з урахуванням необхідності реалізації всіх етапів нечіткого виводу, а сам процес виведення реалізується на основі одного з розглянутих алгоритмів нечіткого виводу.

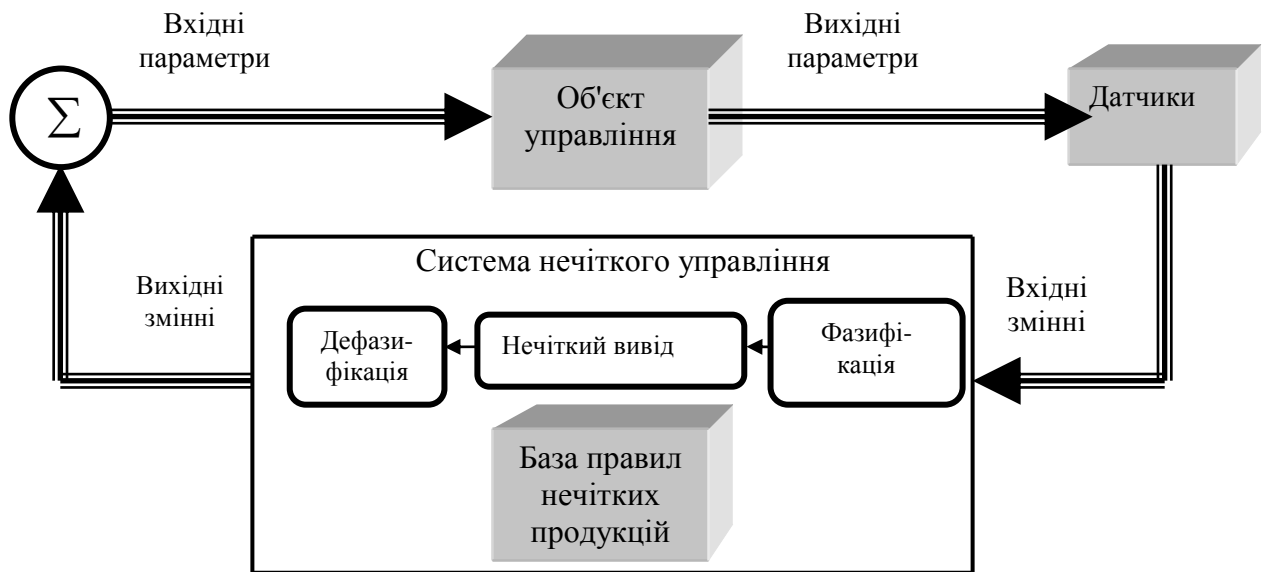


Рис.3.62. Архітектура компонентів процесу нечіткого управління

3.9.1. Приклади розробки нечітких моделей прийняття рішень у середовищі MATLAB

3.9.1.1. Нечітка модель управління кондиціонером повітря в приміщенні

В якості прикладу використання систем нечіткого виводу в задачах управління розглянемо задачу управління кондиціонером повітря в приміщенні. Дана задача ілюструє процес стабілізації температури повітря в приміщенні, в якому встановлено побутовий кондиціонер. Припустимо, що в якості алгоритму нечіткого виводу використовується алгоритм Мамдані.

Змістовна постановка задачі

У приміщенні встановлено побутовий кондиціонер, який дозволяє охолоджувати або нагрівати повітря в цьому приміщенні. Найбільш комфортні умови в приміщенні створюються за деякої стабільної температури повітря. Завдання полягає в тому, щоб зробити регулювання кондиціонера автоматичним, забезпечуючи постійну температуру повітря в приміщенні (рис.3.63).

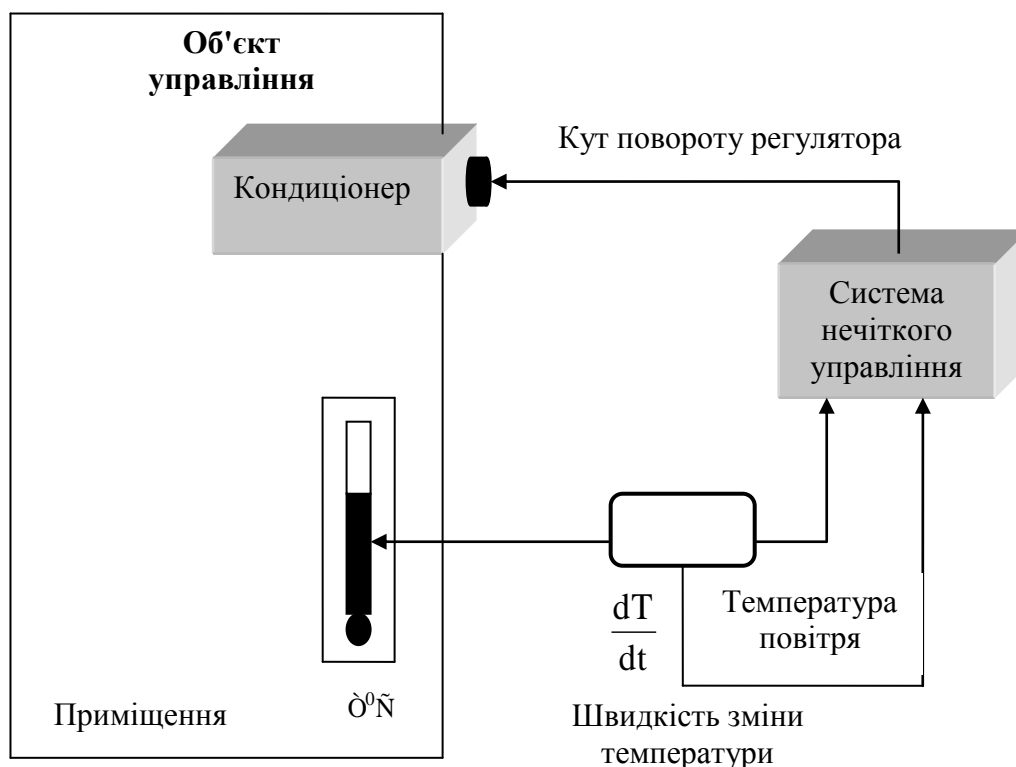


Рис.3.63. Ілюстрація моделі нечіткого управління кондиціонером повітря в приміщенні

Досвід використання кондиціонерів показує, що процес охолодження або нагрівання повітря в приміщенні має певну інерційність: після включення режиму "холод" відбувається нагнітання холодного повітря, у зв'язку з чим температура повітря в приміщенні поступово падає. При цьому у момент вимкнення цього режиму температура продовжує падати протягом невеликого, але кінцевого проміжку часу. Аналогічна картина спостерігається при включенні і відключенні режиму «тепло».

Припустимо, що в розглянутій моделі кондиціонера увімкнення режиму "холод" здійснюється поворотом регулятора ліворуч, увімкнення режиму "тепло" здійснюється поворотом регулятора вправо відносно деякої точки, в якій кондиціонер вимкнений.

Щоб урахувати цю особливість процесу управління кондиціонером і виключити додаткові витрати, пов'язані з частим вмиканням і вимиканням зазначених режимів, необхідно розглядати в якості вихідного параметра не тільки температуру повітря в приміщенні, але і швидкість її зміни. У цьому випадку емпіричні знання про розглянуту проблемну область можуть бути представлені у формі евристичних правил, які застосовуються у разі ручного регулювання температури повітря в приміщенні з кондиціонером:

1. Якщо температура повітря в приміщенні дуже тепла, а швидкість зміни температури позитивна, то слід увімкнути режим "холод", повернувши регулятор кондиціонера на дуже великий кут вліво.

2. Якщо температура повітря в приміщенні дуже тепла, а швидкість зміни температури негативна, то слід увімкнути режим "холод", повернувши регулятор кондиціонера на невеликий кут вліво.

3. Якщо температура повітря в приміщенні тепла, а швидкість зміни температури позитивна, то слід увімкнути режим "холод", повернувши регулятор кондиціонера на великий кут вліво.

4. Якщо температура повітря в приміщенні дуже тепла, а швидкість зміни температури негативна, то кондиціонер варто вимкнути.

5. Якщо температура повітря в приміщенні дуже холодна, а швидкість зміни температури негативна, то слід увімкнути режим "тепло", повернувши регулятор кондиціонера на дуже великий кут вправо.

6. Якщо температура повітря в приміщенні дуже холодна, а швидкість зміни температури позитивна, то слід увімкнути режим "тепло", повернувши регулятор кондиціонера на невеликий кут вправо.

7. Якщо температура повітря в приміщенні холодна, а швидкість зміни температури негативна, то слід увімкнути

режим "тепло", повернувши регулятор кондиціонера на великий кут вправо.

8. Якщо температура повітря в приміщенні холодна, а швидкість зміни температури позитивна, то кондиціонер варто вимкнути.

9. Якщо температура повітря в приміщенні дуже тепла, а швидкість зміни температури дорівнює нулю, то слід включити режим "холод", повернувши регулятор кондиціонера на великий кут вліво.

10. Якщо температура повітря в приміщенні тепла, а швидкість зміни температури дорівнює нулю, то слід увімкнути режим "холод", повернувши регулятор кондиціонера на невеликий кут вліво.

11. Якщо температура повітря в приміщенні дуже холодна, а швидкість зміни температури дорівнює нулю, то слід включити режим "тепло", повернувши регулятор кондиціонера на великий кут вправо.

12. Якщо температура повітря в приміщенні холодна, а швидкість зміни температури дорівнює нулю, то слід увімкнути режим "тепло", повернувши регулятор кондиціонера на невеликий кут вправо.

13. Якщо температура повітря в приміщенні у межах норми, а швидкість зміни температури позитивна, то слід увімкнути режим "холод", повернувши регулятор кондиціонера на невеликий кут вліво.

14. Якщо температура повітря в приміщенні у межах норми, а швидкість зміни температури негативна, то слід увімкнути режим "тепло", повернувши регулятор кондиціонера на невеликий кут вправо.

15. Якщо температура повітря в приміщенні в межах норми, а швидкість зміни температури дорівнює нулю, то кондиціонер варто вимкнути.

Ця інформація використовується при побудові бази правил системи нечіткого виводу, яка дозволяє реалізувати дану модель нечіткого управління.

Побудова бази нечітких лінгвістичних правил

Для формування бази правил систем нечіткого виводу необхідно попередньо визначити вхідні і вихідні лінгвістичні змінні. Очевидно, що в якості однієї з вхідних лінгвістичних змінних слід використовувати температуру повітря в приміщенні: β_1 – "температура повітря", а в якості другої вхідної лінгвістичної змінної β_2 – "швидкість зміни температури повітря". В якості вихідної лінгвістичної змінної будемо використовувати кут повороту регулятора включення режимів "холод" і "тепло" кондиціонера: β_3 – "кут повороту регулятора". Для скорочення запису правил використовуються символічні позначення.

Таблиця 3.2. Загальноприйняті скорочення для основних термів лінгвістичних змінних

Символічне позначення	Україномовна нотація
NB	Негативне велике
NM	Негативне середнє
NS	Негативне мале
ZN	Негативне близьке до нуля
Z	Нуль, близьке до нуля
ZP	Позитивне близьке до нуля
PS	Позитивне мале
PM	Позитивне середнє
PB	Велике позитивне

У цьому випадку система нечіткого виводу буде містити 15 правил нечітких продукцій такого вигляду:

- Правило 1: ЯКЩО " $\beta_1 \in PB$ " І " $\beta_2 \in PS$ " ТО " $\beta_3 \in NB$ "
 Правило 2: ЯКЩО " $\beta_1 \in PB$ " І " $\beta_2 \in NS$ " ТО " $\beta_3 \in NS$ "
 Правило 3: ЯКЩО " $\beta_1 \in PS$ " І " $\beta_2 \in PS$ " ТО " $\beta_3 \in NM$ "

- Правило 4: ЯКЩО " $\beta_1 \in PS$ " І " $\beta_2 \in NS$ " ТО " $\beta_3 \in Z$ "
 Правило 5: ЯКЩО " $\beta_1 \in NB$ " І " $\beta_2 \in NS$ " ТО " $\beta_3 \in PB$ "
 Правило 6: ЯКЩО " $\beta_1 \in NB$ " І " $\beta_2 \in PS$ " ТО " $\beta_3 \in PS$ "
 Правило 7: ЯКЩО " $\beta_1 \in NS$ " І " $\beta_2 \in NS$ " ТО " $\beta_3 \in PM$ "
 Правило 8: ЯКЩО " $\beta_1 \in NS$ " І " $\beta_2 \in PS$ " ТО " $\beta_3 \in Z$ "
 Правило 9: ЯКЩО " $\beta_1 \in PB$ " І " $\beta_2 \in Z$ " ТО " $\beta_3 \in NM$ "
 Правило 10: ЯКЩО " $\beta_1 \in PS$ " І " $\beta_2 \in Z$ " ТО " $\beta_3 \in NS$ "
 Правило 11: ЯКЩО " $\beta_1 \in NB$ " І " $\beta_2 \in Z$ " ТО " $\beta_3 \in PM$ "
 Правило 12: ЯКЩО " $\beta_1 \in NS$ " І " $\beta_2 \in Z$ " ТО " $\beta_3 \in PS$ "
 Правило 13: ЯКЩО " $\beta_1 \in Z$ " І " $\beta_2 \in PS$ " ТО " $\beta_3 \in NS$ "
 Правило 14: ЯКЩО " $\beta_1 \in Z$ " І " $\beta_2 \in NS$ " ТО " $\beta_3 \in PS$ "
 Правило 15: ЯКЩО " $\beta_1 \in Z$ " І " $\beta_2 \in Z$ " ТО " $\beta_3 \in Z$ "

Фазифікація вхідних змінних

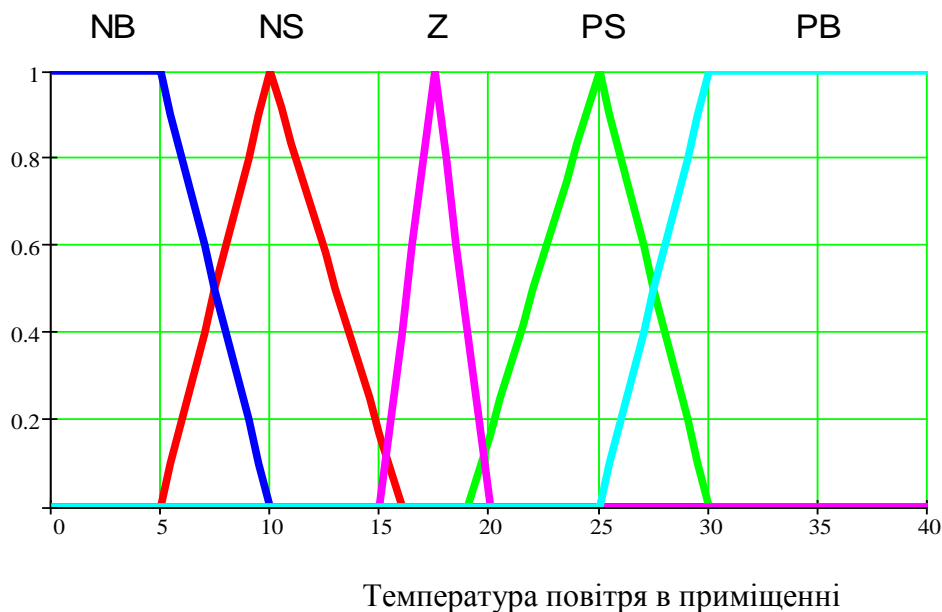


Рис.3.64. Графіки функцій належності для термів вхідної лінгвістичної змінної "Температура повітря"

В якості терм-множини першої вхідної лінгвістичної змінної використовуємо множину: $T_1 = \{ \text{"дуже холодна"}, \text{"холодна"}, \text{"в межах норми"}, \text{"тепла"}, \text{"дуже тепла"} \}$ або в символічному вигляді з функціями приналежності, зображеними на рис.3.64.

В якості терм-множини другої вхідної лінгвістичної змінної використовуємо множину: $T_2 = \{\text{"негативна"}, \text{"дорівнює нулю"}, \text{"позитивна"}\}$ або в символічному вигляді $T_2 = \{NS, Z, PS\}$ з функціями приналежності, зображеними на рис.3.65.

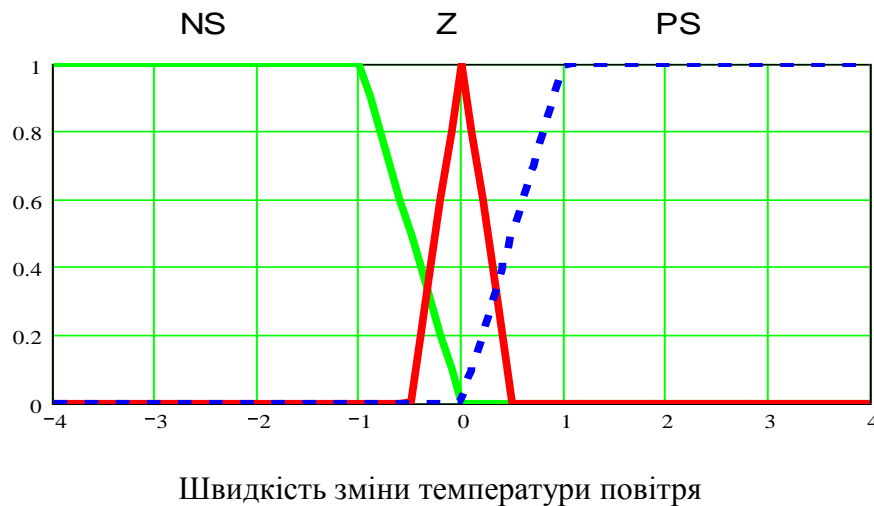


Рис.3.65. Графіки функцій належності для термів вхідної лінгвістичної змінної "Швидкість зміни температури"

В якості терм-множини вихідної лінгвістичної змінної використовуємо множину: $T_3 = \{\text{"дуже великий кут вліво"}, \text{"великий кут вліво"}, \text{"невеликий кут вліво"}, \text{"вимкнути кондиціонер"}, \text{"невеликий кут вправо"}, \text{"великий кут вправо"}, \text{"дуже великий кут вправо"}\}$ або в символічному вигляді $T_3 = \{NB, NM, NS, Z, PS, PM, PB\}$ з функціями приналежності, зображеними на рис. 3.66.

При цьому температура повітря вимірюється в градусах Цельсія, швидкість зміни температури повітря в градусах Цельсія на хвилину, а кут повороту – в кутових градусах. В останньому випадку поворот регулятора вправо означає включення режиму «тепло» і позитивний напрямок відліку, а поворот уліво – включення режиму "холод" і негативний напрямок відліку. Використовуючи в якості алгоритму виводу алгоритм Мамдані, розглянемо приклад для випадку, коли поточна температура

повітря дорівнює $20\text{ }^{\circ}\text{C}$, а швидкість її зміни позитивна і дорівнює $0,2\text{ }^{\circ}\text{C}/\text{хв}$. В цьому випадку фазифікація першої вхідної лінгвістичної змінної приводить до значення ступеня інтенсивності $0,15$ для терма PS, а фазифікація другої нечіткої змінної приводить до значення істинності $0,5$ для терма Z і значенням $0,2$ для терма PS. Відповідні підумови використовуються в правилах нечітких продукцій з номерами 3 і 10. Ці правила вважаються активними і використовуються в поточному процесі нечіткого виводу.

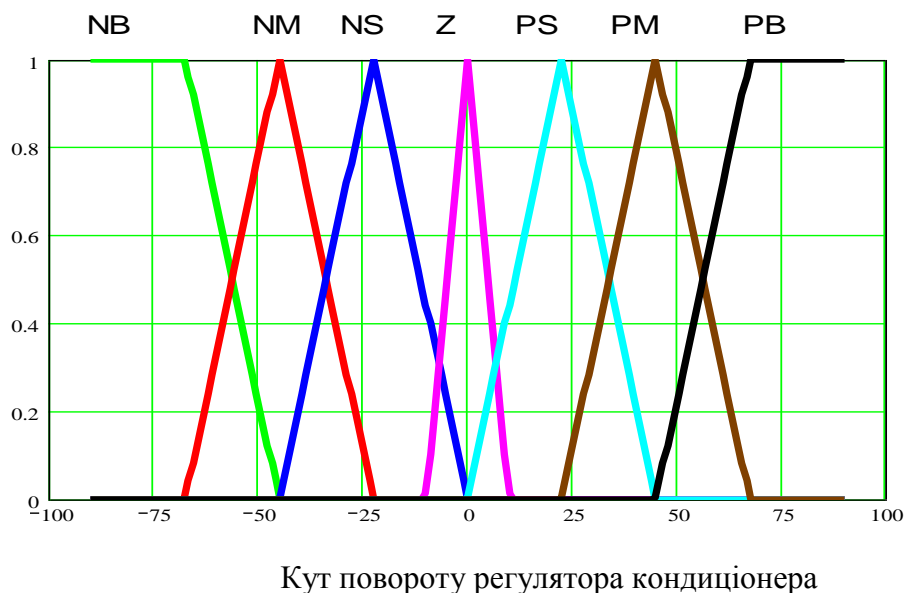


Рис.3.66. Графіки функцій належності для термів вихідної лінгвістичної змінної "Кут повороту регулятора"

Агрегування підумов правила 3 дає в результаті число $0,15$, агрегування підумов правила 10 – також число $0,15$. Таким етапом нечіткого виводу є активізація виводів нечітких правил продукцій. Оскільки всі виводи правил 1–5 задані у формі нечітких лінгвістичних висловлювань першого виду, а вагові коефіцієнти правил за замовчуванням дорівнюють 1, то активація правил 3 і 10 приводить до двох нечітких множин. Акумулявання правил для 3 і 10 висновків нечітких правил продукцій проводиться з використанням операції max-диз'юнкції.

Дефазифікація вихідної лінгвістичної змінної "Кут повороту регулятора" методом центру площі призводить до значення керуючої змінної, рівного повороту регулятора кондиціонера вліво на кут (наближене значення). Це значення відповідає увімкненню режиму «холод» на третину своєї потужності та є результатом рішення нечіткого виводу.

Нечітка модель управління кондиціонером повітря в приміщенні

Розробку нечіткої моделі будемо виконувати з використанням графічних засобів системи MATLAB. З цією метою відкриємо редактор FIS і визначимо 2 вхідні змінні з іменами "температура" (β_1) і "швидкість" (β_2) та одну вихідну змінну з ім'ям "кут" (β_3).

Вигляд графічного інтерфейсу редактора FIS для цих змінних показаний на рис.3.67.

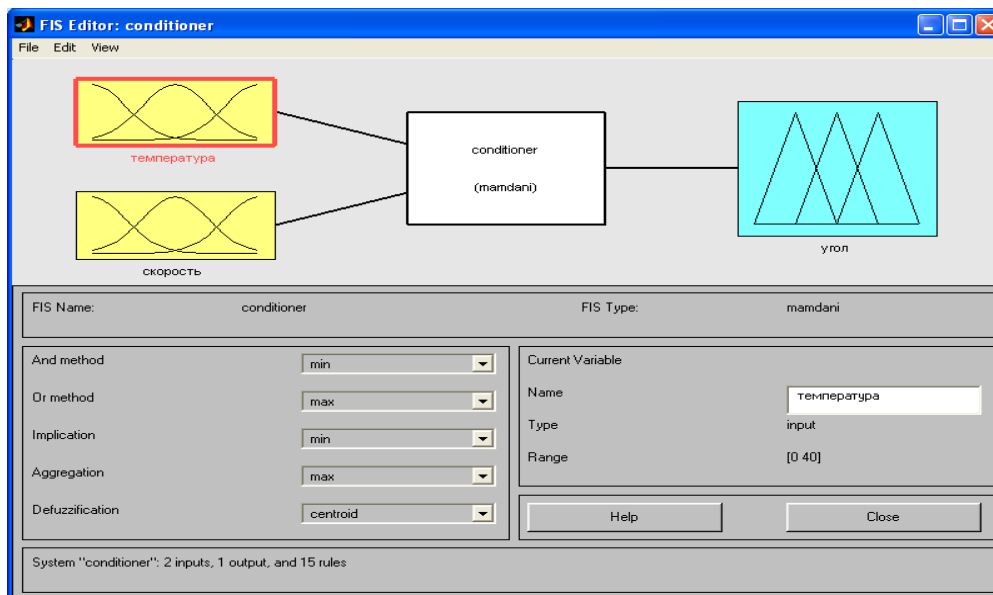


Рис.3.67. Графічний інтерфейс редактора FIS після визначення вхідних і вихідних змінних розроблюваної системи нечіткого виводу

Оскільки ми використовуємо систему нечіткого виводу за алгоритмом Мамдані, залишимо без зміни тип, запропонований

системою MATLAB за замовчуванням. Немає необхідності змінювати й інші параметри розроблюваної нечіткої моделі, запропоновані системою MATLAB за замовчуванням, такі як логічні операції (**min** – для нечіткого логічного І, **max** – для нечіткого логічного АБО), методи імплікації (**min**), агрегування (**max**) і дефазифікації (**centroid**).

Далі слід визначити функції належності термів для кожної зі змінних системи нечіткого виводу. Для цієї мети скористаємося редактором функцій приналежності системи MATLAB. Для першої вхідної змінної слід додати два додаткових терми до трьох, заданих за замовчуванням, і визначити параметри відповідних функцій приналежності.

Чисельні значення цих параметрів такі NB – 0,5,10; NS – 5,10,16; Z – 15,17.5,20; PS – 19,25,30; PB – 25,30. Вигляд графічного редактора функцій приналежності після завдання першої вхідної змінної "температура" для системи нечіткого виводу показаний на рис. 3.68. Для другої вхідної змінної слід залишити три терми, задані за замовчуванням, і змінити тільки тип і параметри функцій приналежності.

Чисельні значення цих параметрів такі NS – 1,0; Z – 0.5,0,0.5; PS – 0,1. Для вихідної змінної слід додати чотири терми до трьох, заданих за замовчуванням, і задати параметри відповідних функцій приналежності. Чисельні значення цих параметрів такі NB – 67.5,-45; NM – 67.5,-45,-22.5; NS – 45,-22.5,0; Z – -10,0,10; PS – 0,22.5,45; PM – 22.5,45,67.5; PB – 45,67.5.

Вигляд графічного інтерфейсу редактора функцій приналежності після визначення вихідної змінної «кут» для системи нечіткого виводу показаний на рис.3.69.

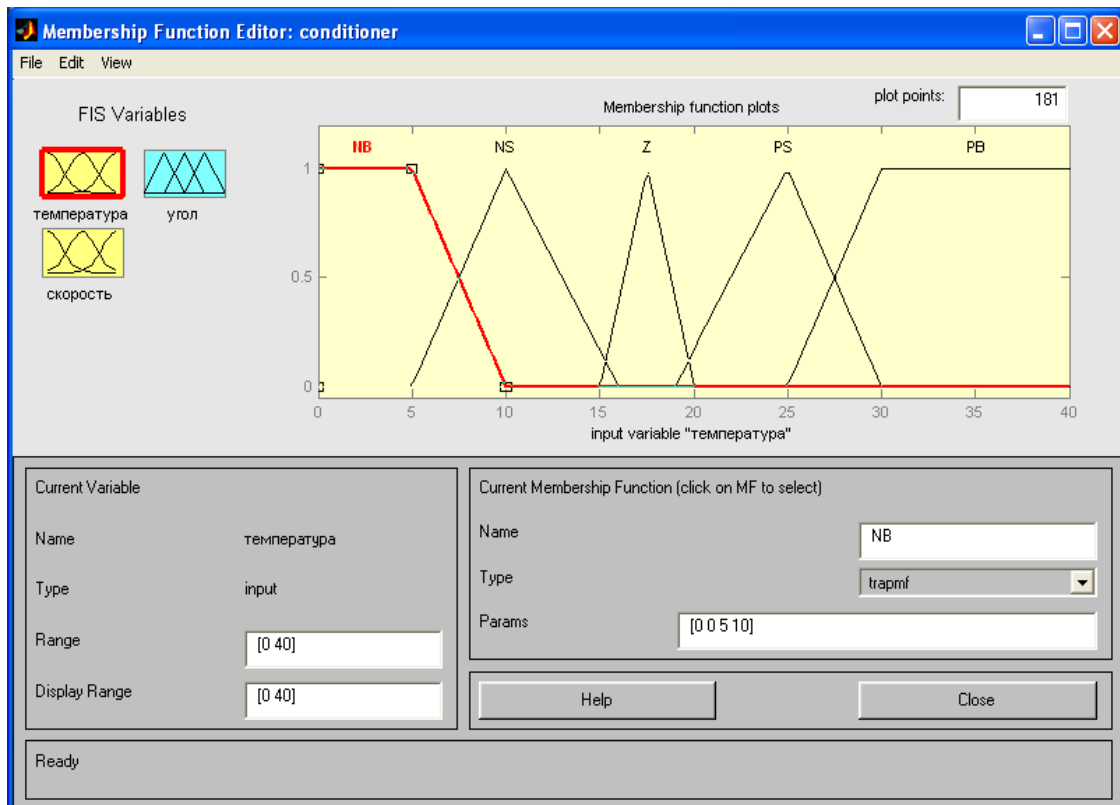


Рис.3.68. Графічний інтерфейс редактора функцій приналежності після завдання першої вхідної змінної "температура" для системи нечіткого виводу "кондиціонер"

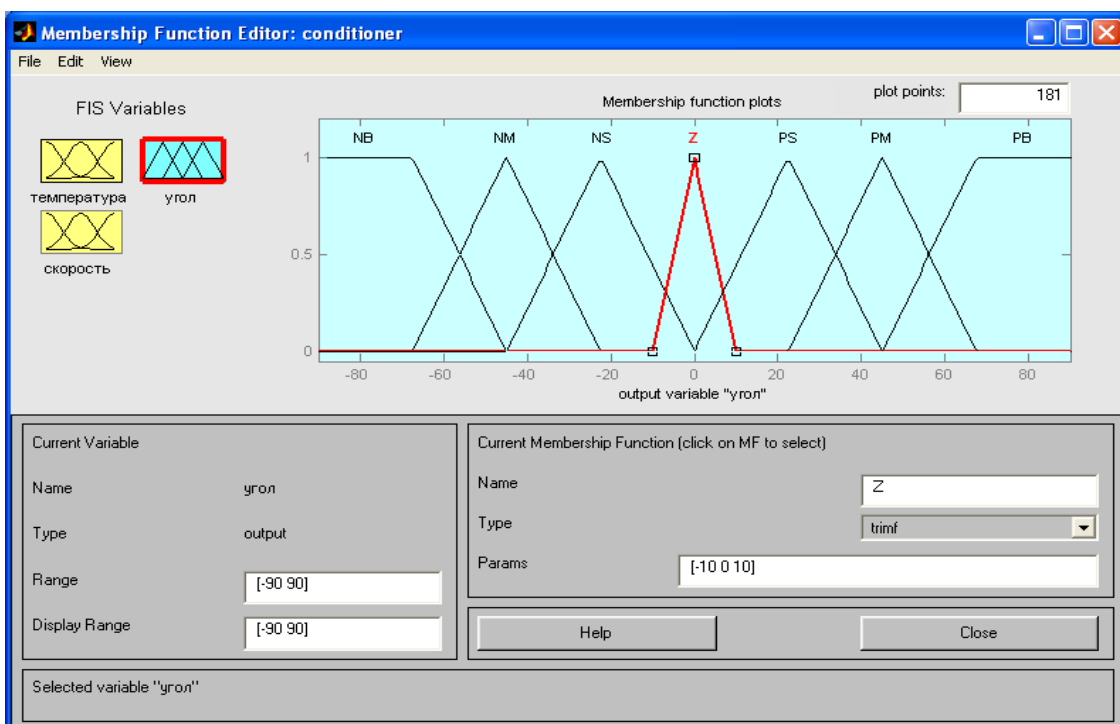


Рис.3.69. Графічний інтерфейс редактора функцій приналежності після визначення вихідної змінної "кут" для системи нечіткого виводу "conditioner"

Далі поставимо 15 правил для розроблюваної системи нечіткого виводу. Для цієї мети скористаємося редактором правил системи MATLAB. Вигляд графічного інтерфейсу редактора правил після введення всіх 15 правил нечіткого висновку показаний на рис.3.71.

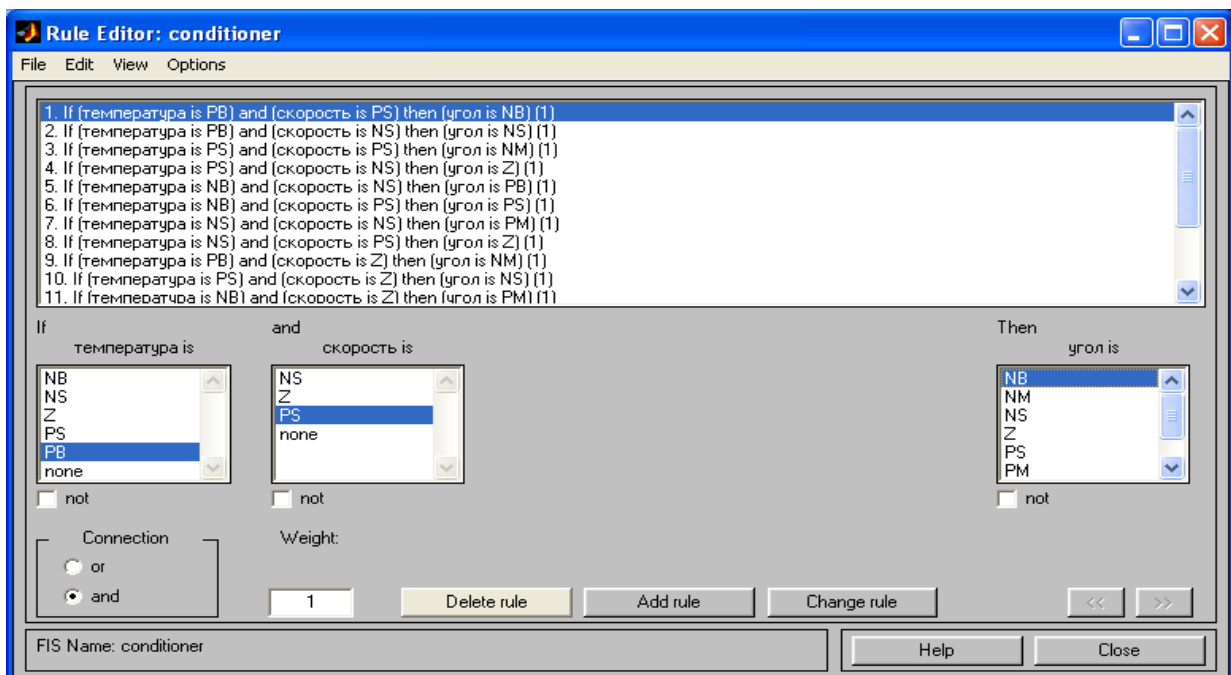


Рис.3.71.Графічний інтерфейс редактора правил після визначення бази правил системи нечіткого виводу "conditioner"

Потім можна виконати оцінку побудованої системи нечіткого виводу для задачі автоматичного управління кондиціонером у приміщенні. З цією метою відкриємо програму перегляду правил системи MATLAB і введемо значення вхідних змінних для окремого випадку, коли поточна температура повітря в приміщенні дорівнює 20°C , а швидкість її зміни позитивна і становить $0,2^{\circ}\text{C}/\text{хв}$. Процедура нечіткого виводу, виконана системою MATLAB для розробленої нечіткої моделі, видає в результаті значення вихідної змінної "кут", дорівнює $-33,8^{\circ}\text{C}$ (рис. 3.71).

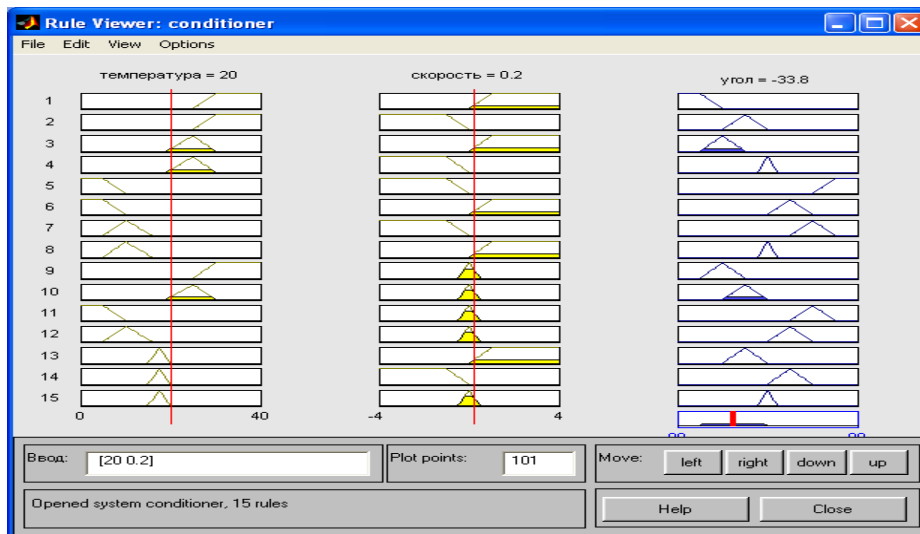


Рис.3.71. Графічний інтерфейс програми перегляду правил після виконання процедури нечіткого виводу для значень вхідних змінних [200.2]

Дане значення відповідає увімкненню режиму "холод" кондиціонера на третину своєї потужності.

Процес аналізу та дослідження побудованої нечіткої моделі включає в себе виконання нечітких висновків для різних значень вхідних змінних й оцінки отриманих результатів з метою встановлення адекватності моделі і внесення в неї необхідних змін у разі неузгодженості окремих результатів.

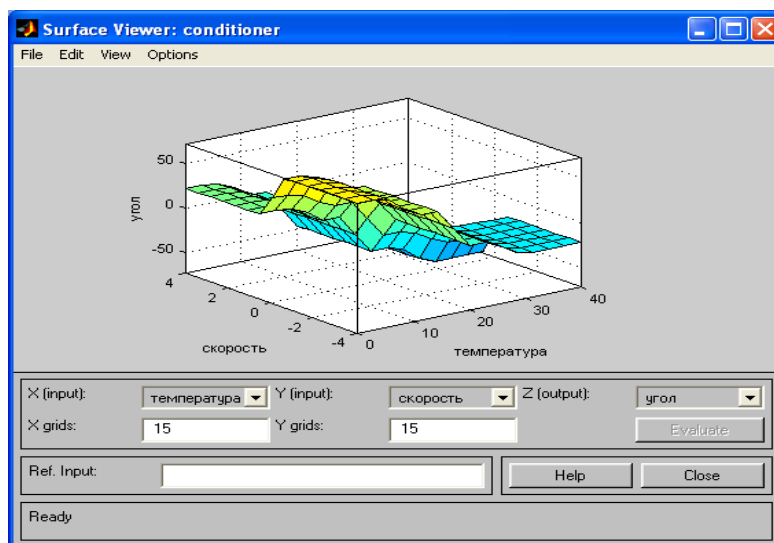


Рис.3.72. Візуалізація поверхні нечіткого виводу для системи нечіткого виводу "conditioner"

Для загального аналізу розробленої нечіткої моделі може виявитися корисною візуалізація відповідної поверхні нечіткого виводу (рис.3.72). Дана поверхня нечіткого висновку дозволяє встановити залежність значень вихідної змінної від значень вхідних змінних нечіткої моделі системи управління кондиціонером. Ця залежність може послужити основою для програмування контролера або апаратної реалізації відповідного нечіткого алгоритму управління у формі таблиці рішень. На додаток до цього встановлення цієї залежності є, по суті, розв'язанням задачі, відомої в класичній теорії управління як задача синтезу керуючих впливів.

3.9.1.2. Оцінювання фінансової спроможності клієнтів при наданні банківських кредитів

У даний час у сфері бізнесу та економіки широко використовуються людино-машинні системи, призначені для обробки нечітких даних, що реалізують і надають допомогу прийнятті економічних рішень, маркетинг, поради щодо вкладення капіталу, різного роду управління і планування, аналіз ризику, допомога у підготовці контрактів і т. ін.

В якості прикладу наведемо розв'язання задачі оцінювання фінансової спроможності клієнта з метою прийняття рішень при наданні довгострокових кредитів на будівництво під заставу нерухомості з допомогою нечіткої моделі. При видачі довгострокових кредитів на будівництво будинків або котеджів під заставу нерухомості для оцінки спроможності клієнтів банками традиційно використовується метод експертних оцінок. При цьому метою банків є отримання максимального прибутку від укладених угод по наданню кредитів і виключення можливості фінансових втрат. Тому інтереси банків зосереджені, з одного боку, на збільшенні кількості успішних угод, а з іншого боку, на запобіганні невдалих угод, коли клієнт не повертає виданий кредит або повертає його невчасно. Традиційно

підставою для прийняття рішень з надання кредитів у майбутньому служить досвід успішних угод, укладених у минулому.

Аналіз стратегії надання кредитів на будівництво будівель показує, що для оцінювання фінансової спроможності клієнтів можуть бути використані різні характеристики, такі як місце розташування будівлі, що будується, якість передбачуваного виконання опоряджувальних робіт, оцінка активів потенційного клієнта, оцінка доходу потенційного клієнта за мінусом фіксованих витрат, величина відсотків за кредитом. При цьому власне фінансова спроможність клієнта оцінюється його кредитоспроможністю.

Однією з перших формальних моделей, запропонованих для вирішення даної задачі, була статистична модель, заснована на ймовірнісній інтерпретації кількісної оцінки позитивного рішення про надання кредиту. Більш детальний аналіз цієї моделі з часом показав її неадекватність, пов'язану з недостатнім обсягом статистичної вибірки і змінюваними з плином часу умовами надання кредитів. Саме з цієї причини була запропонована ідея розробки нечіткої моделі для оцінювання фінансової спроможності клієнтів з метою прийняття рішень про надання довгострокових кредитів. У якості нечіткої моделі використовується система нечіткого виводу з наступними вхідними і вихідними змінними.

У нечіткій моделі використані п'ять вхідних змінних (оцінка місця розташування об'єкта, що будується, якість оздоблювальних робіт, оцінка активів, оцінка доходу клієнта, розмір належних до сплати відсотків) і одна вихідна – оцінка кредитоспроможності, яка є основою для прийняття рішення керівництвом банку. При цьому рішення про надання кредиту приймається керівництвом банку лише в разі високої оцінки цієї вихідної змінної.

В якості **першої** вхідної змінної використовується оцінка місця розташування будівлі, що будується, яка безпосередньо оцінює проект будівлі, що будується, беручи до уваги розміщення будівлі в тому чи іншому конкретному районі міста або приміському регіоні. Очевидно, чим вища ця оцінка, тим більш ліквідним виглядає проект у випадку його реалізації на ринку нерухомості.

В якості **другої** вхідної змінної використовується якість передбачуваного виконання опоряджувальних робіт згідно з архітектурним проектом будівлі, що будується. Ця змінна вносить додатковий елемент в оцінку вартості будівлі, що будується.

В якості **третьої** вхідної змінної використовується оцінка активів, яка використовується для оцінки майна або авуарів у разі неспроможності потенційного клієнта при неповерненні їм взятого кредиту. Дійсно, величина наданого кредиту повинна ґрунтуватися не тільки на обліку вартості споруджуваного будинку, але і на власній капіталізації клієнта.

В якості **четвертої** вхідної змінної використовується оцінка доходу потенційного клієнта за мінусом фіксованих витрат, яка також використовується в разі неспроможності потенційного клієнта при неповерненні їм взятого кредиту. Чим вище значення цієї змінної, тим більш успішним видається надання кредиту клієнту.

В якості **п'ятої** вхідної змінної використовується величина відсотків, що підлягають сплаті відповідно до передбачуваного плану виплат за отриманим кредитом. Ця змінна пов'язана з терміном надання кредиту та його величиною, дозволяючи об'єднати в собі відповідні характеристики кредиту. Чим вища величина виплат по відсотках, тим вищими мають бути значення активів і доходів для позитивного рішення про надання кредиту потенційному клієнту.

В якості **вихідної** змінної використовується оцінка кредитоспроможності, яка є основою для прийняття рішень керівництвом банку з надання кредиту потенційним клієнтам. При цьому рішення про надання кредиту приймається керівництвом банку лише в разі високої оцінки цієї вихідної змінної.

Аналіз надання кредитів на будівництво будівель показує, що для аналізу фінансової заможності клієнтів керівництво банку приймає такі евристичні правила:

1. Якщо величина доходу низька і середня величина виплат, то кредитоспроможність дуже низька.

2. Якщо величина доходу низька і величина виплат висока, то кредитоспроможність дуже низька.

3. Якщо середня величина доходу і величина виплат висока, то кредитоспроможність дуже низька.

4. Якщо активи низькі і величина доходу низька, то кредитоспроможність дуже низька.

5. Якщо активи низькі і середня величина доходу, то кредитоспроможність дуже низька.

6. Якщо активи середні і величина доходу низька, то кредитоспроможність дуже низька.

7. Якщо якість обробки погана, активи низькі і величина доходу висока, то кредитоспроможність дуже низька.

8. Якщо якість обробки погана, активи середні і середня величина доходу, то кредитоспроможність дуже низька.

9. Якщо якість обробки погана, активи високі і величина доходу низька, то кредитоспроможність дуже низька.

10. Якщо якість обробки погана, активи високі і величина доходу середня, то кредитоспроможність дуже низька.

11. Якщо непрестижне місце розташування, якість обробки хороша, активи низькі і величина доходу висока, то кредитоспроможність середня.

12. Якщо непрестижне місце розташування, якість обробки прекрасна, активи низькі і величина доходу висока, то кредитоспроможність середня.

13. Якщо престижне місце розташування, якість обробки хороша, активи низькі і величина доходу висока, то кредитоспроможність середня.

14. Якщо дуже престижне місце розташування, якість обробки хороша, активи низькі і величина доходу висока, то кредитоспроможність середня.

15. Якщо непрестижне місце розташування, якість обробки хороша, активи середні і величина доходу середня, то кредитоспроможність середня.

16. Якщо непрестижне місце розташування, якість обробки прекрасна, активи середні і середня величина доходу, то кредитоспроможність середня.

17. Якщо престижне місце розташування, якість обробки хороша, активи середні і середня величина доходу, то кредитоспроможність середня.

18. Якщо дуже престижне місце розташування, якість обробки хороша, активи середні і середня величина доходу, то кредитоспроможність середня.

19. Якщо непрестижне місце розташування, якість обробки хороша, активи високі і величина доходу низька, то кредитоспроможність середня.

20. Якщо непрестижне місце розташування, якість обробки прекрасна, активи високі і величина доходу низька, то кредитоспроможність середня.

21. Якщо престижне місце розташування, якість обробки хороша, активи високі і величина доходу низька, то кредитоспроможність середня.

22. Якщо дуже престижне місце розташування, якість обробки хороша, активи високі і величина доходу середня, то кредитоспроможність середня.

23. Якщо непрестижне місце розташування, якість обробки хороша, активи високі і середня величина доходу, то кредитоспроможність середня.

24. Якщо непрестижне місце розташування, якість обробки прекрасна, активи високі і середня величина доходу, то кредитоспроможність середня.

25. Якщо престижне місце розташування, якість обробки хороша, активи високі і середня величина доходу, то кредитоспроможність середня.

26. Якщо дуже престижне місце розташування, якість обробки хороша, активи високі і середня величина доходу, то кредитоспроможність середня.

27. Якщо престижне місце розташування, якість оздоблення прекрасне, активи середні і величина доходу висока, то кредитоспроможність дуже висока.

28. Якщо престижне місце розташування, якість обробки прекрасна, активи високі і величина доходу висока, то кредитоспроможність дуже висока.

29. Якщо дуже престижне місце розташування, якість обробки прекрасна, активи середні і величина доходу висока, то кредитоспроможність дуже висока.

30. Якщо дуже престижне місце розташування, якість обробки прекрасна, активи високі і величина доходу висока, то кредитоспроможність дуже висока.

31. Якщо непрестижне місце розташування, якість обробки хороша, активи середні і величина доходу висока, то висока кредитоспроможність.

32. Якщо непрестижне місце розташування, якість обробки прекрасна, активи середні і величина доходу висока, то висока кредитоспроможність.

33. Якщо престижне місце розташування, якість обробки хороша, активи середні і величина доходу висока, то висока кредитоспроможність.

34. Якщо дуже престижне місце розташування, якість обробки хороша, активи середні і величина доходу висока, то висока кредитоспроможність.

35. Якщо непрестижне місце розташування, якість обробки хороша, активи високі і величина доходу висока, то висока кредитоспроможність.

36. Якщо непрестижне місце розташування, якість обробки прекрасна, активи високі і величина доходу висока, то висока кредитоспроможність.

37. Якщо престижне місце розташування, якість обробки хороша, активи високі і величина доходу висока, то висока кредитоспроможність.

38. Якщо дуже престижне місце розташування, якість обробки хороша, активи високі і величина доходу висока, то висока кредитоспроможність.

39. Якщо престижне місце розташування, якість обробки прекрасна, то висока кредитоспроможність.

40. Якщо дуже престижне місце розташування, якість обробки прекрасна, то висока кредитоспроможність.

Нечітка модель оцінювання фінансової спроможності клієнтів

У базі правил системи нечіткого виводу використовувалося 40 правил нечітких продукцій. При побудові нечіткої моделі оцінки фінансової спроможності потенційних клієнтів було зроблено припущення про те, що всі розглянуті змінні вимірюються в балах в інтервалі дійсних чисел від 0 до 10. При цьому найнижча оцінка значення кожної з змінних є 0, а найвищою – 10.

Фазифікація вхідних і вихідних змінних

Як терм-множину першої вхідної змінної "Місцезнаходження" будемо використовувати множину:

$T_1 = \{ \text{"непрестижне"}, \text{"престижне"}, \text{"дуже престижне"} \}$
або в символічному вигляді $T_1 = \{PS, PM, PB\}$ з функціями належності термів, зображеними на рис. 3.73 а (PS: 0; 0; 0,5; 6; PM: 1,5; 3,5; 6; 9; PB: 6; 9; 10; 10).

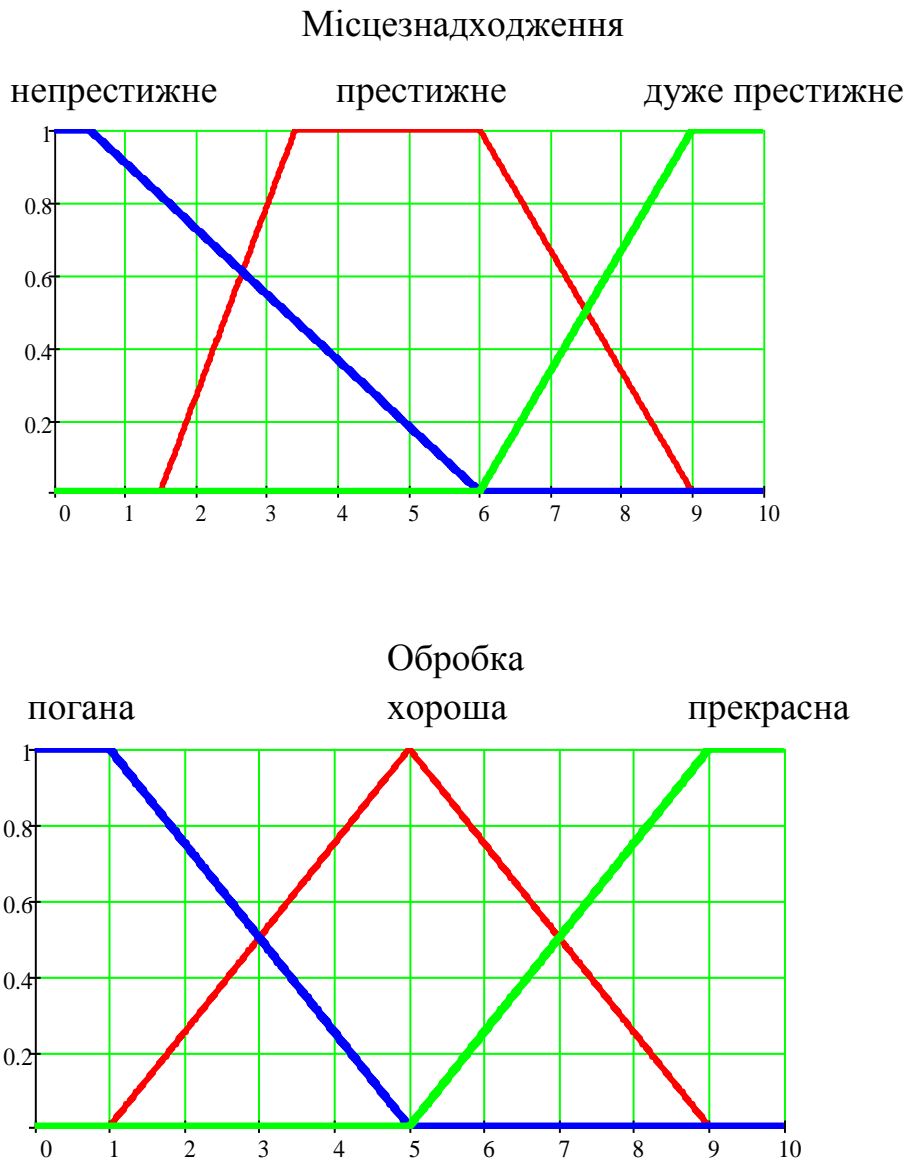


Рис.3.73. Графіки функцій належності для термів лінгвістичних змінних "Місцезнаходження" та "Обробка"

В якості терм-множини другої вхідної змінної «Обробка» будемо використовувати аналогічну множину:

$$T_2 = \{ \text{"погана"}, \text{"хороша"}, \text{"прекрасна"} \}$$

або в символному вигляді $T_2 = \{PS, PM, PB\}$ з функціями належності термів (PS: 0; 0; 1; 5; PM: 1; 5; 9; PB: 5; 9; 10; 10).

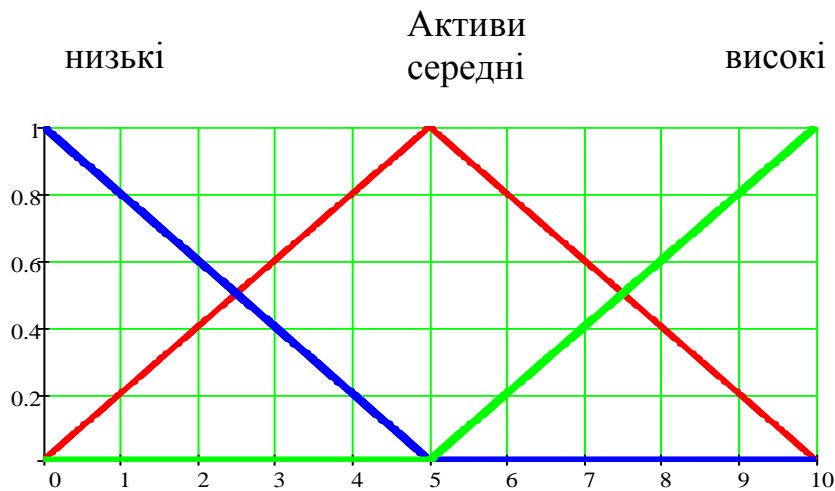


Рис.3.74. Графік функцій приналежності для терма лінгвістичної змінної "Активи", що вимірюється в балах

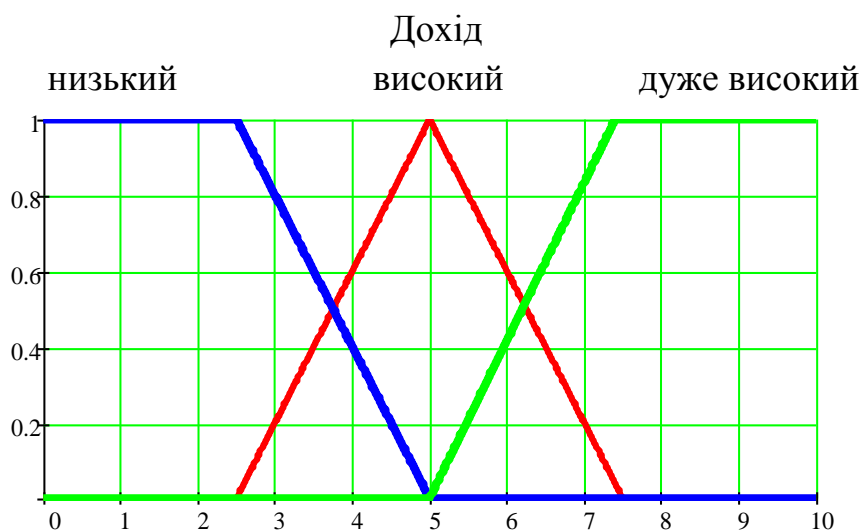


Рис.3.75. Графік функцій приналежності для терма лінгвістичної змінної "Дохід", яка вимірюється в балах

В якості терм-множини третьої лінгвістичної змінної "Активи" будемо використовувати множину:

$$T_3 = \{ \text{"низькі"}, \text{"середні"}, \text{"високі"} \}$$

або в символному вигляді $T_3 = \{PS, PM, PB\}$ з функціями належності термів, зображених на рис. 3.74 (PS: 0; 0; 5; PM: 0; 5; 10; PB: 5; 10; 10).

В якості терм-множини четвертої лінгвістичної змінної "Дохід" будемо використовувати множину:

$$T_4 = \{\text{"низький"}, \text{"середній"}, \text{"високий"}\}$$

або в символному вигляді $T_4 = \{PS, PM, PB\}$ з функціями належності термів, зображених на рис.3.75 (PS: 0; 0; 2,5; 5; PM: 2,5; 5; 7,5; PB: 5;7,5; 10; 10).

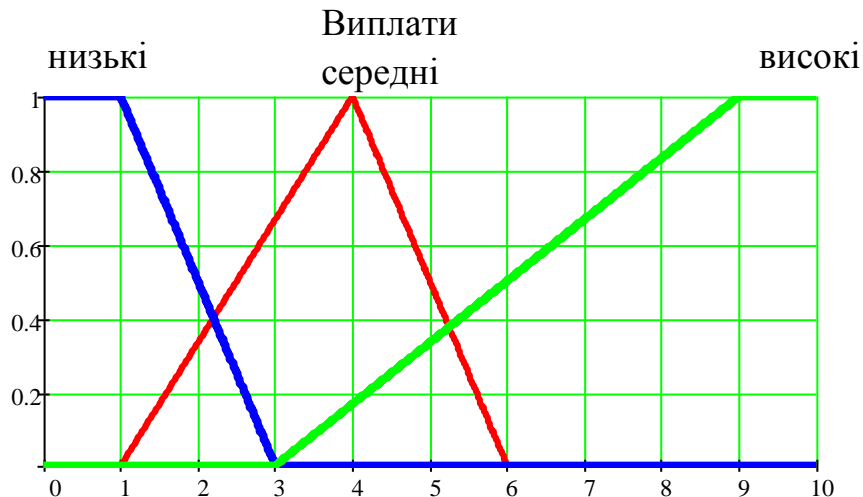


Рис.3.76. Графік функцій приналежності для терма лінгвістичної змінної "Виплати", яка вимірюється в балах

В якості терм-множини п'ятої лінгвістичної змінної "Виплати" будемо використовувати множину:

$T_5 = \{\text{"низькі"}, \text{"середні"}, \text{"високі"}\}$ або в символному вигляді $T_5 = \{PS, PM, PB\}$ з функціями належності термів, зображених на рис. 3.76 (PS: 0; 0; 1; 3; PM: 1; 4; 6; PB: 3; 9; 10;10).

В якості терм-множини вихідної лінгвістичної змінної "Кредитоспроможність" будемо використовувати множину:

$$T_6 = \{\text{"дуже низька"}, \text{"низька"}, \text{"середня"}, \text{"висока"}, \text{"дуже висока"}\}$$

або в символному вигляді $T_6 = \{NB, NS, Z, PS, PB\}$ з функціями належності термів, зображених на рис. 3.77 (NB: 0; 0; 3,33; NS: 1,66; 3,33; 5; Z: 3,33; 5; 6,66; PS: 5; 6,66; 8,33; PB: 6,66; 10; 10).

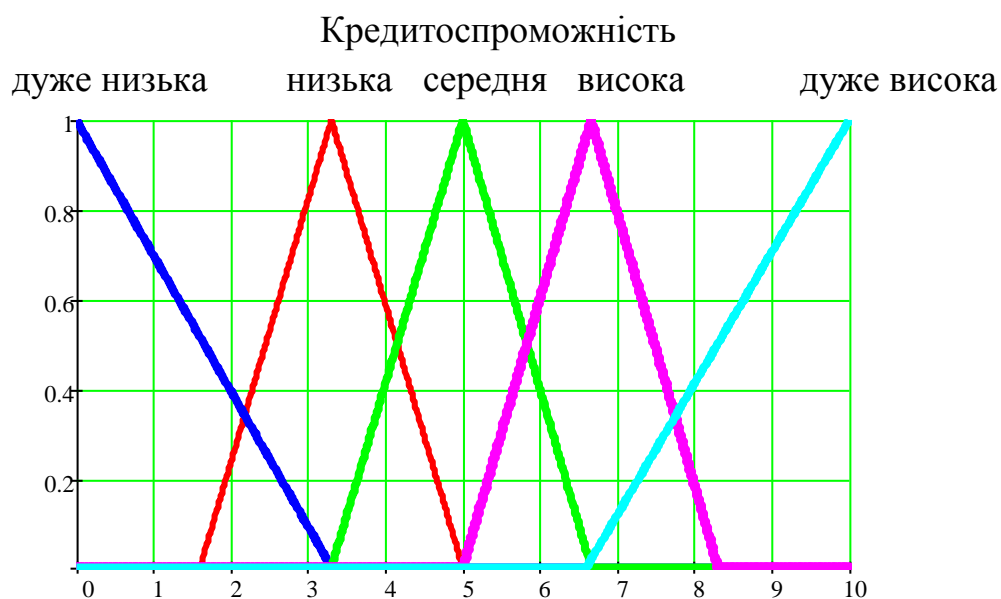


Рис.3.77. Графік функцій приналежності для терма лінгвістичної змінної "Кредитоспроможність", що вимірюється в балах

Формування бази правил систем нечіткого виводу

Таким етапом побудови моделі є побудова бази правил. Для цього будемо використовувати 40 правил нечітких продукцій, які зручно представити у вигляді табл. 3.3.

В якості схеми нечіткого виводу будемо використовувати метод Мамдані, тому методом активізації буде MIN, який розраховується за формулою:

\min – активізація

$$\mu'(y) = \min\{c_i, \mu(y)\},$$

де $\mu(y)$ – функція приналежності терма, який є деякою вихідною змінною ω_i , заданою на множині Y .

Таблиця 3.3. Правила нечітких продукцій для розглянутої системи нечіткого виводу

Номер правила	Розташування	Обробка	Активи	Дохід	Виплати	Кредитоспроможність
1				PS	PM	NB
2				PS	PB	NB
3				PM	PB	NB
4			PS	PS		NB
5			PS	PM		NB
6			PM	PS		NB
7		PS	PS	PB		NB
8		PS	PM	PM		NB
9		PS	PB	PS		NB
10		PS	PB	PM		Z
11	PS	PM	PS	PB		Z
12	PS	PB	PS	PB		Z
13	PM	PM	PS	PB		Z
14	PB	PM	PS	PB		Z
15	PS	PM	PM	PM		Z
16	PS	PB	PM	PM		Z
17	PM	PM	PM	PM		Z
18	PB	PM	PM	PM		Z
19	PS	PM	PB	PS		Z
20	PS	PB	PB	PS		Z
21	PM	PM	PB	PS		Z
22	PB	PM	PB	PS		Z
23	PS	PM	PB	PM		Z
24	PS	PB	PB	PM		Z
25	PM	PM	PB	PM		Z
26	PB	PM	PB	PM		Z
27	PM	PB	PM	PB		PB
28	PM	PB	PB	PB		PB
29	PB	PB	PM	PB		PB
30	PB	PB	PB	PB		PB
31	PS	PM	PM	PB		PS
32	PS	PB	PM	PB		PS
33	PM	PM	PM	PB		PS
34	PB	PM	PM	PB		PS
35	PS	PM	PB	PB		PS
36	PS	PB	PB	PB		PS
37	PM	PM	PB	PB		PS
38	PB	PM	PB	PB		PS
39	PM	PB				PS
40	PB	PB	PM	PB		PS

Далі необхідно визначити методи агрегування підумов. Оскільки у всіх правилах 1–40 в якості логічного зв'язку для підумов застосовуються тільки нечітка кон'юнкція (операція «І»), то в якості методу агрегування будемо використовувати метод *max-диз'юнкції*, який також застосовується у випадку схеми нечіткого виведення методом Мамдані. В якості методу дефазифікації будемо використовувати метод центру тяжіння.

Побудова нечіткої моделі засобами Fuzzy Logic Toolbox та аналіз отриманих результатів

Побудову нечіткої моделі буде здійснено засобами fuzzy logic toolbox системи Matlab. З цією метою в редакторі FIS визначимо 5 вхідних змінних з іменами "місцезнаходження", "обробка", "активи", "дохід", "виплати", і одну вихідну змінну з ім'ям "кредитоспроможність". Вигляд графічного інтерфейсу редактора FIS для цих змінних показаний на рис. 3.78.

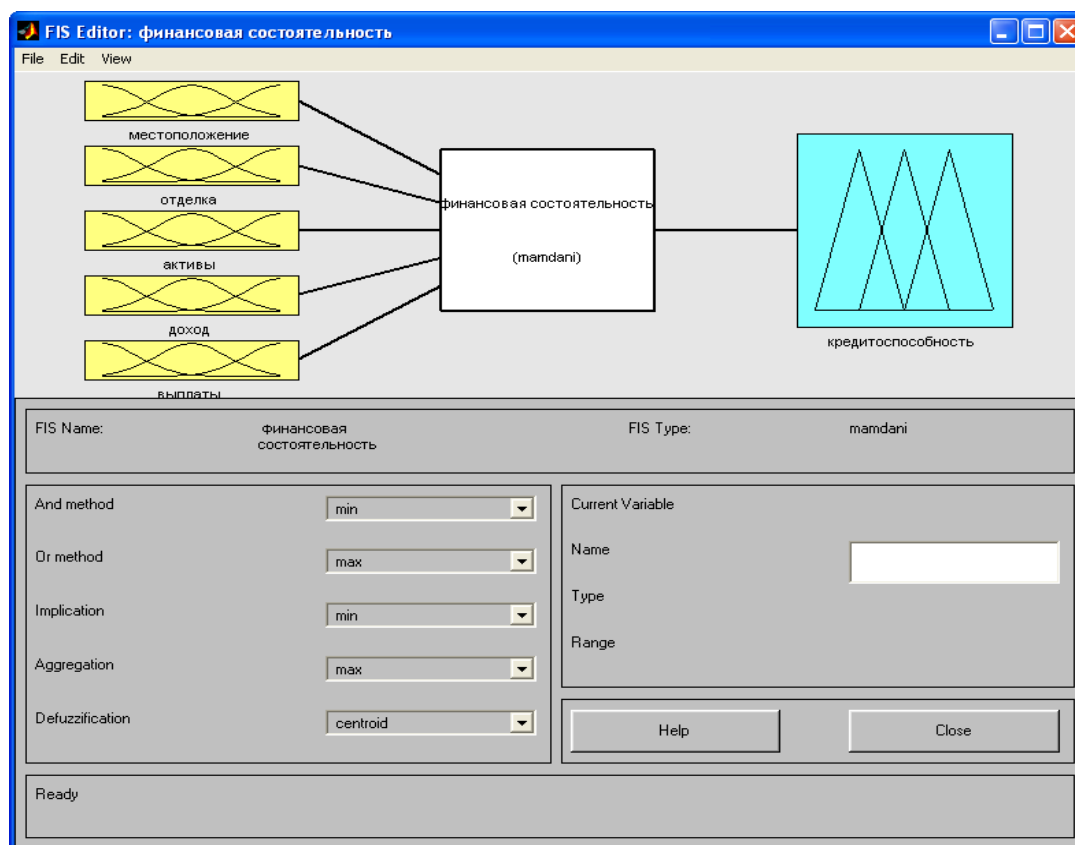


Рис.3.78. Графічний інтерфейс редактора FIS після визначення вхідних та вихідної змінних системи нечіткого виводу фінансової спроможності

Для вирішення поставленої задачі нечіткого моделювання будемо використовувати систему нечіткого виведення Мамдані. Залишимо без зміни параметри розроблюваної нечіткої моделі, запропоновані системою MATLAB за замовчуванням, а саме: логічні операції (\min – для нечіткого логічного І, \max – для нечіткого логічного АБО), методи імплікації (\min), агрегування (\max) і дефазифікації (centroid).

Далі слід визначити функції належності термів для кожної з 5 вхідних і єдиної вихідної змінної розглянутої системи нечіткого виводу. Для цього скористаємося редактором функцій приналежності системи MATLAB. Будемо використовувати типи функцій приналежності та відповідні чисельні значення їх параметрів, які зображені на рис. 3.77. Графічний інтерфейс редактора функцій приналежності для вихідної змінної «кредитоспроможність» зображений на рис. 3.79.

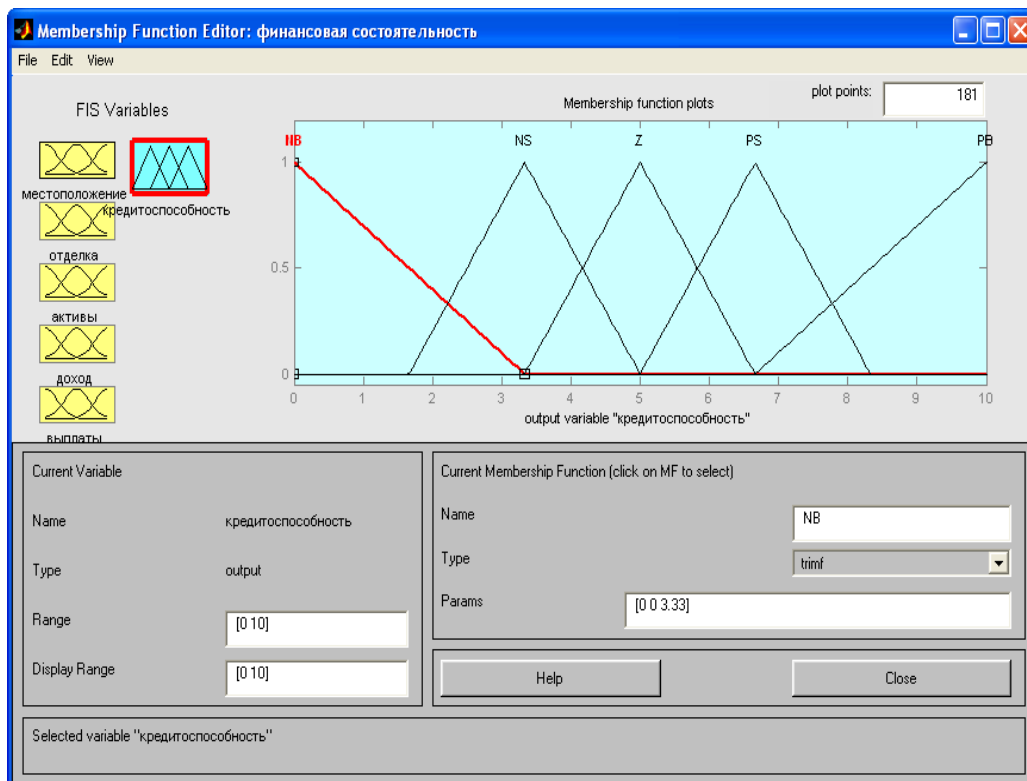


Рис.3.79. Графічний інтерфейс редактора функцій приналежності для вихідної змінної "кредитоспроможність"

Далі поставимо 40 правил для розроблюваної системи нечіткого виводу (таблиця правил). Для цієї мети скористаємося редактором правил системи MATLAB. Вид графічного інтерфейсу редактора правил після завдання всіх 40 правил нечіткого виводу показаний на рис.3.80. Так як у робочому вікні відображаються всі змінні нечіткої моделі, для управління режимом відображення змінних правил слід скористатися спеціальними кнопками >> і <<, розташованими у нижній частині редактора правил.

Тепер можна виконати аналіз побудованої системи нечіткого виводу для розглянутої задачі оцінки фінансової спроможності клієнтів. З цією метою відкриємо вікно перегляду правил системи MATLAB і введемо значення вхідних змінних для окремого випадку, коли значення вхідної змінної «місцезнаходження» оцінюється у 8 балів, значення вхідної змінної «обробка» також оцінюється у 8 балів, значення вхідної змінної «активи» оцінюється у 9 балів, значення вхідної змінної «дохід» оцінюється у 9 балів, і нарешті, значення вхідної змінної «виплати» оцінюється у 5 балів. Це досить високі оцінки вхідних змінних, які навіть на інтуїтивному рівні свідчать на користь відповідного клієнта.

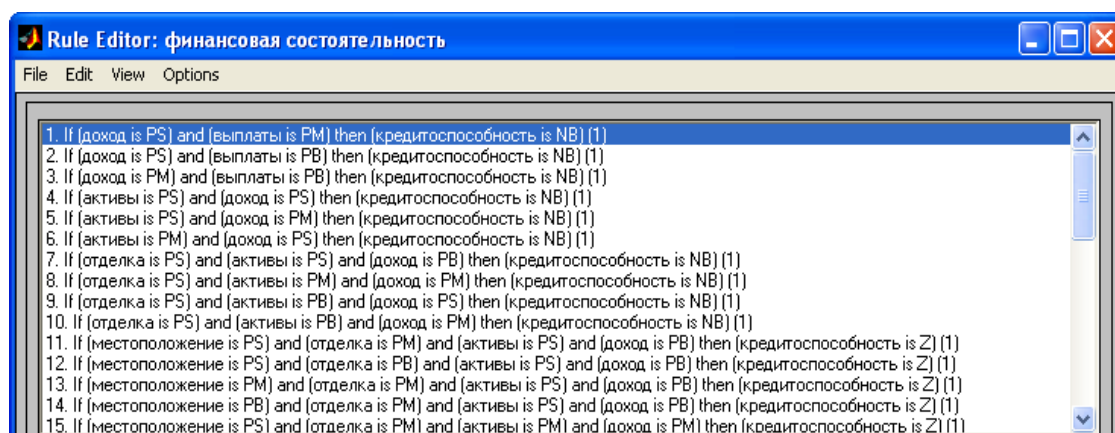


Рис.3.80. Графічний інтерфейс редактора правил після визначення бази правил системи нечіткого виводу фінансової спроможності

Процедура нечіткого виводу, виконана системою MATLAB для розробленої нечіткої моделі, видає в результаті значення вихідної змінної "кредитоспроможність", рівне 7,77 бала (рис. 3.81). Це досить висока оцінка фінансової спроможності потенційного клієнта, яке може служити підставою для позитивного рішення з боку банку про надання кредиту під заставу.

Виконаємо аналіз побудованої системи нечіткого виводу для іншого варіанту вихідних даних з більш низькими оцінками значень вхідних змінних, а саме 3,5,4,5,2. Рис. 3.82. ілюструє результат вихідної змінної "кредитоспроможність" рівний 4,01 бала.

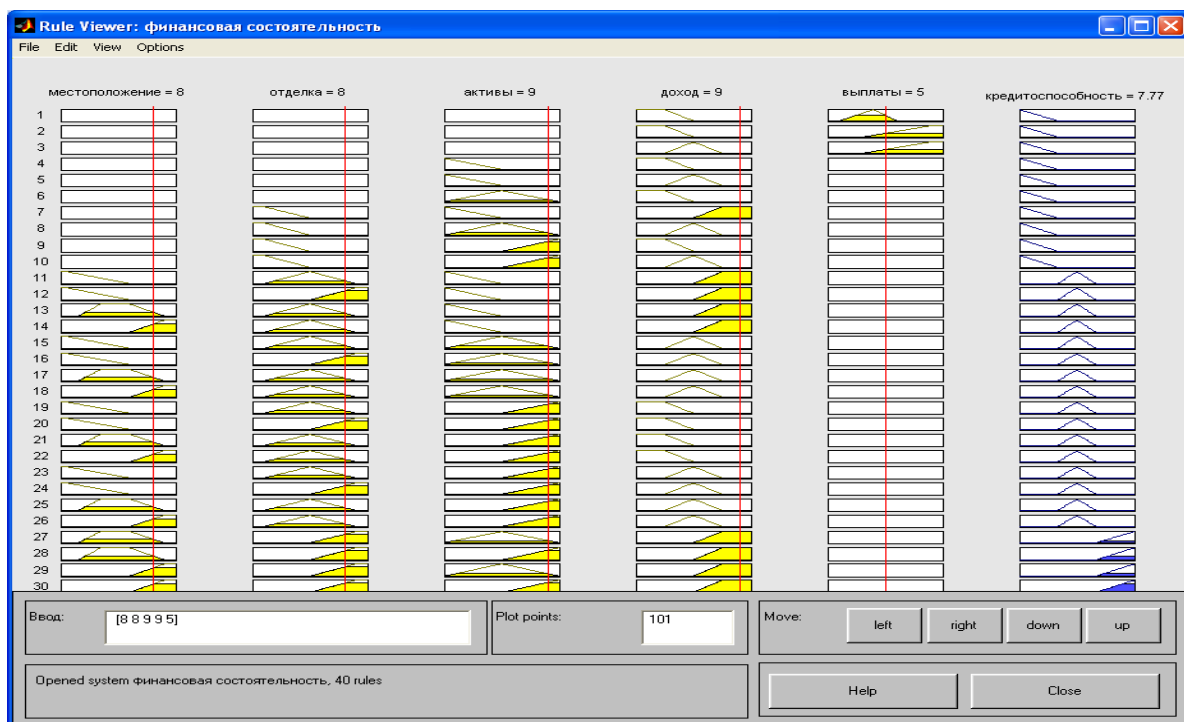


Рис.3.81. Графічний інтерфейс програми перегляду правил після виконання процедури нечіткого виводу для значень вхідних змінних 8, 8, 9, 9, 5

Це досить низька оцінка фінансової спроможності потенційного клієнта, яка може служити підставою для негативного рішення про надання кредиту.

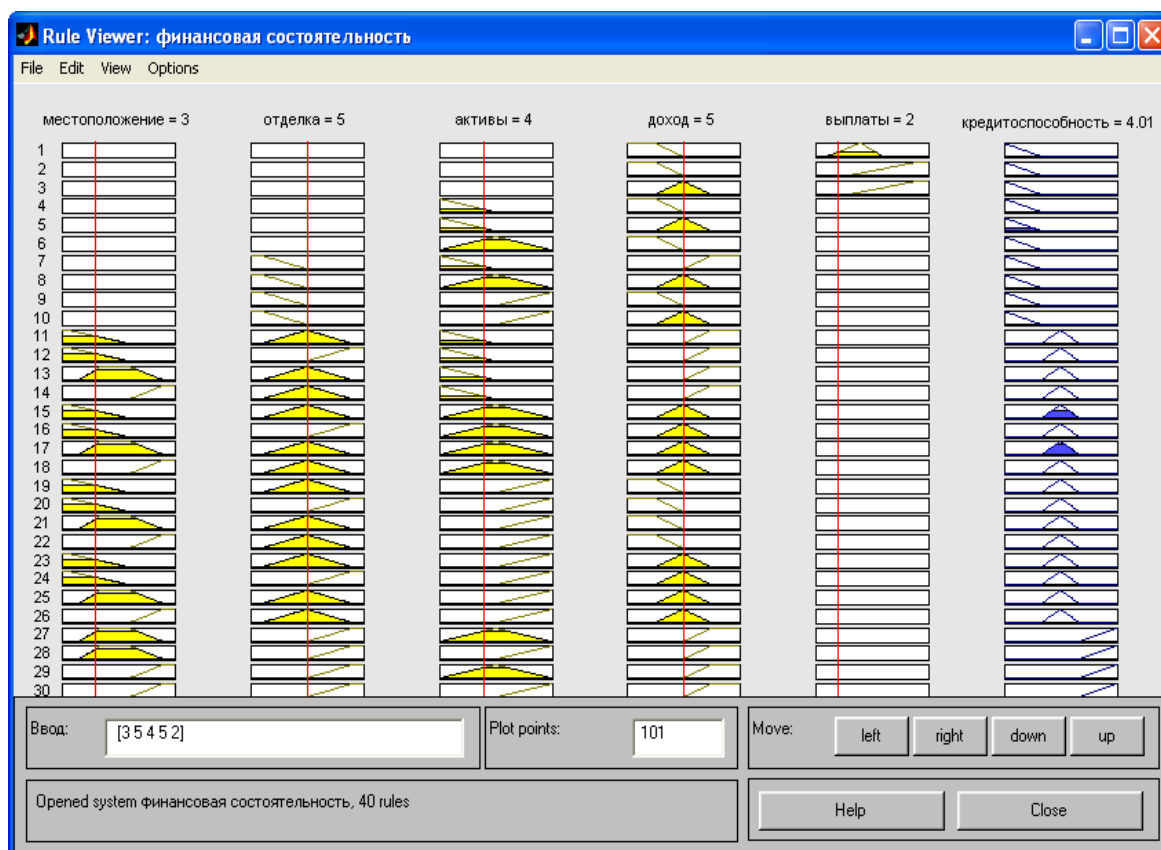


Рис.3.82. Графічний інтерфейс програми перегляду правил після виконання процедури нечіткого виводу для значень вхідних змінних 3, 5, 4, 5, 2

Порівняння результатів нечіткого виводу для двох розглянутих варіантів значень вхідних змінних показує, що граничне значення вихідної змінної "кредитоспроможність", яке впливає на рішення про надання кредиту, може бути обрано не більше 5 балів.

Для загального аналізу розробленої нечіткої моделі може виявитися корисною візуалізація відповідної поверхні нечіткого виводу (рис.3.83). Дана поверхня нечіткого висновку дозволяє встановити залежність значень вихідної змінної від значень окремих вхідних змінних нечіткої моделі. Аналіз цих залежностей може служити підставою для зміни функцій належності вхідних змінних або нечітких правил з метою підвищення адекватності системи нечіткого виводу для конкретних стратегій банків.

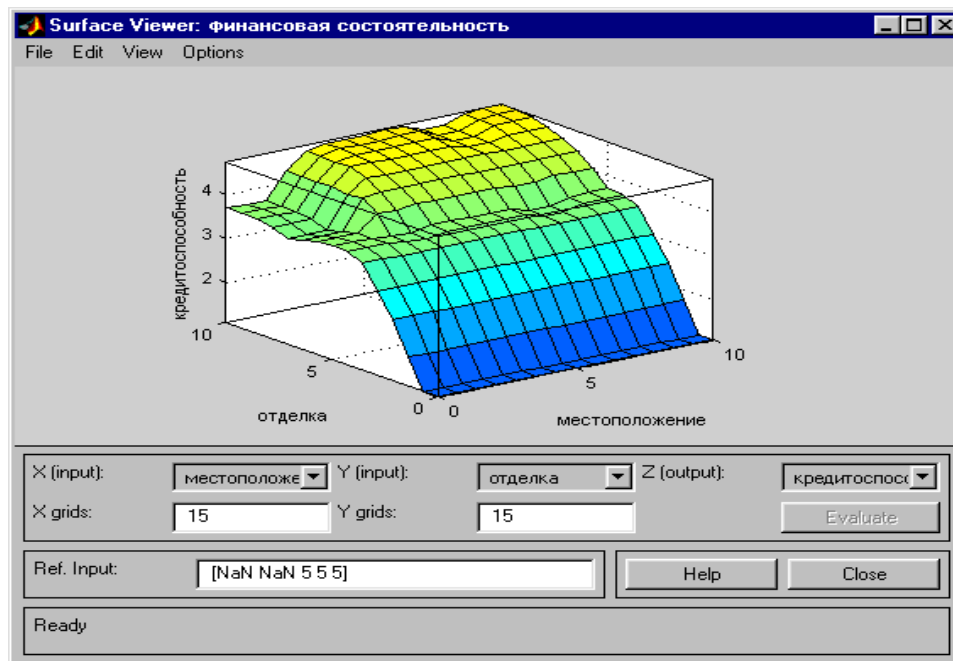


Рис.3.83. Поверхня нечіткого виводу розглянутої моделі для вхідних змінних "місцезнаходження" та "обробка"

Висновок. Системи з нечіткою логікою доцільно застосовувати для складних процесів, коли немає простої математичної моделі; якщо експертні знання про об'єкт чи процес можна сформулювати тільки у лінгвістичній формі. Основні недоліки систем з нечіткою логікою пов'язані з тим, що:

- вихідний набір постульованих нечітких правил формулюється експертом-людиною і може виявитися неповним або суперечливим;
- вид і параметри функцій приналежності, що описують вхідні і вихідні змінні системи, вибираються суб'єктивно і можуть виявитися такими, що не цілком відображають реальну дійсність.

3.9.1.3. Аналіз ризиків інформаційної безпеки

Темпи розвитку сучасних інформаційних технологій значно випереджають темпи розробки рекомендаційної і нормативно-правової бази керівних документів. Виникає необхідність у

вирішенні таких питань: відповідно до яких критеріїв і показників здійснювати оцінку ефективності системи захисту інформації, як забезпечити оцінку і моніторинг інформаційних ризиків в організаціях, особливо малого і середнього бізнесу.

Сучасні методики управління ризиками для аналізу кожного виду ризику використовують ймовірність реалізації загроз і збиток від негативних наслідків, але реально оцінити ймовірність реалізації загроз і ступінь шкоди, що завдається, важко. У більшості випадків експерти в області ІБ, ґрунтуючись на власному досвіді, виконують оцінку у вигляді словесних формулювань, які потім пов'язують з числовими значеннями. Такий механізм отримання оцінок ризиків обмежує можливості методики у цілому, так як упевненість у запропонованій експертом оцінці може носити дискусійний характер.

Для усунення недоліків методик аналізу та оцінки ризиків ІБ пропонується використовувати нечітку логіку, застосування якої ефективно у таких випадках:

- недостатність знань про досліджувану систему;
- неможливість отримання необхідного обсягу інформації;
- інформація заснована на експертних даних, вхідні дані некоректно представлені або не є достатньо точними.

Експертам в області інформаційної безпеки складно дати точну кількісну оцінку компонентам системи забезпечення ІБ організації, таким, наприклад, як "низький рівень організаційної захисту", "середній рівень програмно-апаратного захисту", "висока очевидність ризику" і т. ін. Тому необхідно розглядати ці компоненти з точки зору нечітких множин і лінгвістичних змінних. Використовуючи нечітку логіку для оцінки ризиків ІБ, можна отримати як якісні (виражені у вигляді нечітких понять), так і кількісні характеристики.

Для створення методики оцінки ризиків необхідно розробити експертну систему, яка була б реалізована у вигляді системи нечіткого виведення і дозволяла б визначати величину

ризик на основі суб'єктивних оцінок усіх рівнів інформаційної безпеки. Для моделювання експертної системи використовувався програмний інструментарій MATLAB – високорівнева мова та інтерактивне середовище для програмування чисельних розрахунків і візуалізації результатів, а також Fuzzy Logic Toolbox – пакет розширення MATLAB, що містить інструменти для проектування систем нечіткої логіки.

Таблиця 3.4. Вхідні змінні і відповідні їм терм-множини

Позначення	Найменування	Вид терм-множини
<i>X1</i>	<i>Програмно-апаратний рівень захисту(ПаЗ)</i>	<i>H – задовільна для забезпечення початкового рівня захисту; C – остатня для базової інформаційного захисту; B – високий рівень забезпечення ІБ.</i>
<i>X2</i>	<i>Рівень організаційно-правового захисту (ОргЗ)</i>	<i>H – задовільна для забезпечення початкового рівня захисту; C – остатня для базової інформаційного захисту; B – високий рівень забезпечення ІБ.</i>
<i>X3</i>	<i>Рівень інженерно-технічного захисту (ІнжЗ)</i>	<i>H – задовільна для забезпечення початкового рівня захисту; C – остатня для базової інформаційного захисту; B – високий рівень забезпечення ІБ.</i>

В якості вхідних змінних використовуються визначені експертним шляхом рівні інформаційної безпеки, представлені в таблиці 3.4.

За вихідні лінгвістичні змінні будемо приймати:

Y1 – ризик порушення конфіденційності інформації;

Y2 – ризик порушення цілісності інформації;

Y3 – ризик порушення доступності інформації.

Після визначення вхідних і вихідних змінних введена система нечіткого виведення в інтерактивному режимі, для цього використовувався редактор систем нечіткого виводу FIS (рис. 3.84).

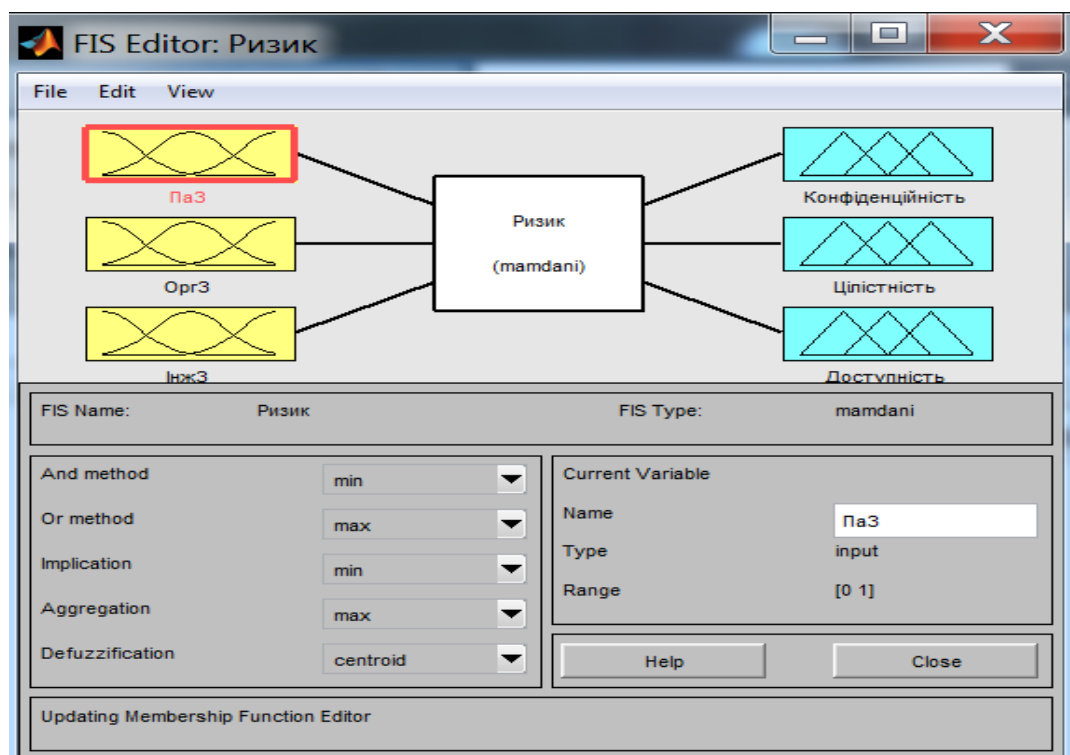


Рис.3.84. Редактор системи нечіткого виводу

Вхідні змінні:

Input1 – ПаЗ;

Input2 – ОргЗ;

Input3 – ІнжЗ.

Вихідні змінні:

Output1– конфіденційність (Y1);

Output2 – цілісність (Y2);

Output3 – доступність (Y3).

Далі визначаються терми та їх функції приналежності для вхідних і вихідних змінних системи нечіткого виведення. Для цього слід скористатися редактором функцій приналежності (рис. 3.85).

У роботі використовуються трапецеїдальна і трикутна форми для завдання функції приналежності.

Для завдання трапецеїдальної функції приналежності необхідна четвірка чисел (a, b, c, d) , її значення в точці x обчислюється відповідно до виразу:

$$MF(x) = \begin{cases} 1 - \frac{b-x}{b-a}, & a \leq x \leq b \\ 1, & b \leq x \leq c \\ 1 - \frac{x-c}{c-b}, & c \leq x \leq d \\ 0 & \text{в інших випадках} \end{cases}$$

При $(b - a) = (d - c)$ трапецеїдальна функція приналежності бере симетричний вид.

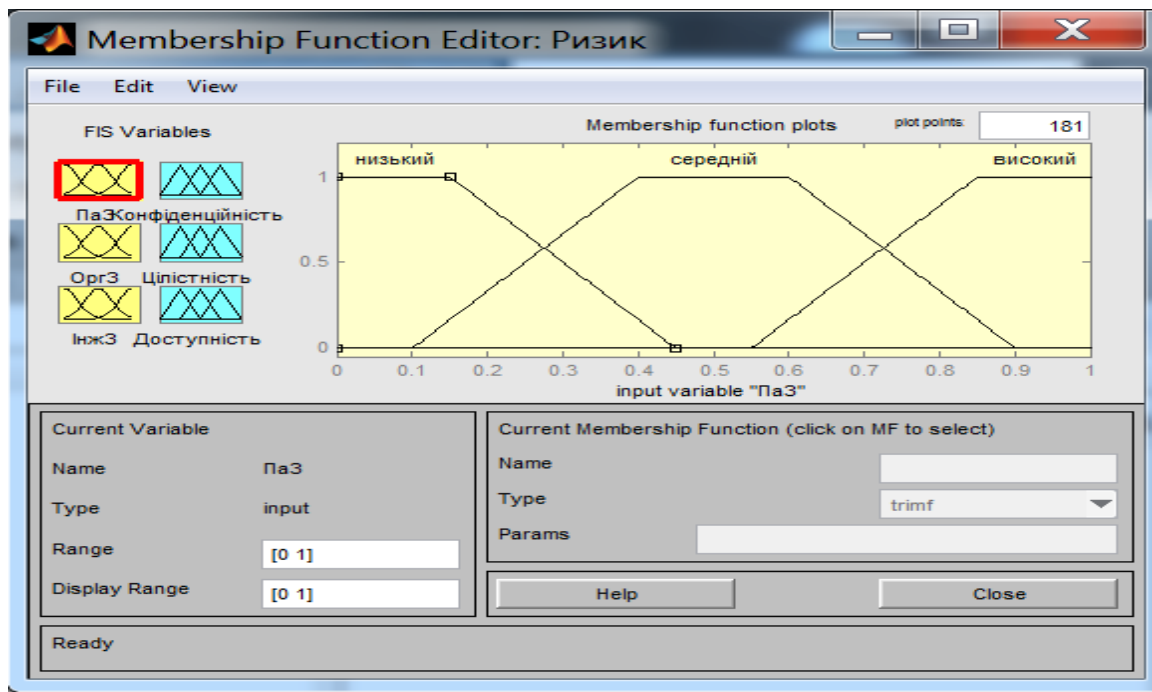


Рис. 3.85. Редактор функцій приналежності для вхідної змінної ПАЗ

Трикутна функція приналежності визначається трійкою чисел (a, b, c) , і її значення в точці x обчислюється відповідно до виразу:

$$MF(x) = \begin{cases} 1 - \frac{b-x}{b-a}, & a \leq x \leq b \\ 1 - \frac{x-b}{c-b}, & b \leq x \leq c \\ 0 & \text{в інших випадках} \end{cases}$$

При $(b - a) = (c - b)$ маємо випадок симетричної трикутної функції приналежності, яка може бути однозначно задана двома параметрами з трійки (a, b, c) .

Для вхідних змінних ПАЗ і ІнжЗ, параметри кожного з термів будуть визначені в такий спосіб (функції приналежності приймаємо як трапецеїдальні): для терма "низький" задамо параметри $[0 \ 0 \ 0,15 \ 0,45]$, для терма "середній" $[0,1 \ 0,4 \ 0,6 \ 0,9]$, для терма "високий" $[0,55 \ 0,85 \ 1 \ 1]$ (рис. 3.86).

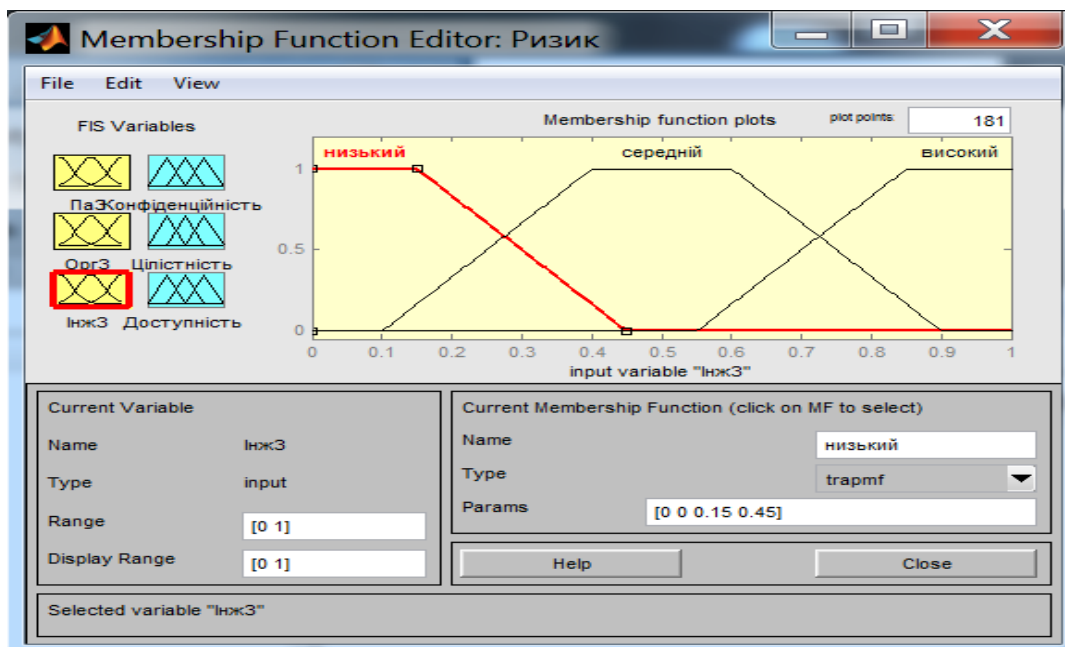


Рис. 3.86. Редактор функцій приналежності для вхідної змінної ІнжЗ

Для вхідної змінної ОргЗ, функції приналежності є трикутними (рис. 3.87).

Низький – $[0 \ 0 \ 0,4]$;

Середній – $[0,1 \ 0,5 \ 0,9]$;

Високий – $[0,6 \ 1 \ 1]$.

На наступному етапі визначаються терм-множини для вихідних змінних. Для вихідних змінних: "конфіденційність, цілісність, доступність", параметри термів будуть такими (рис. 3.88).

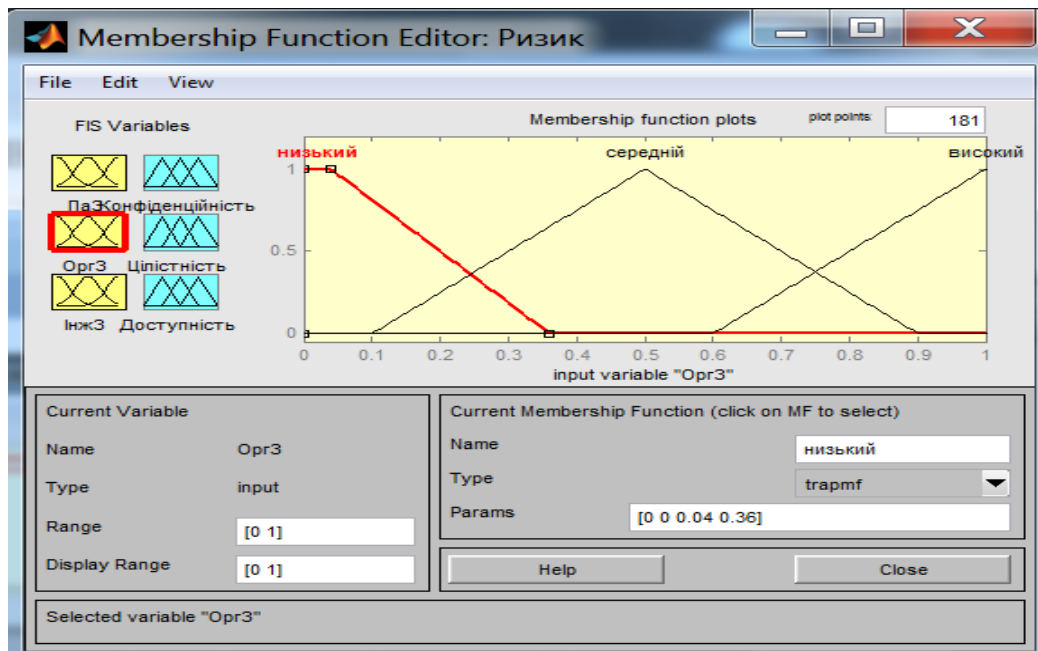


Рис. 3.87. Параметри вхідної змінної Орг3

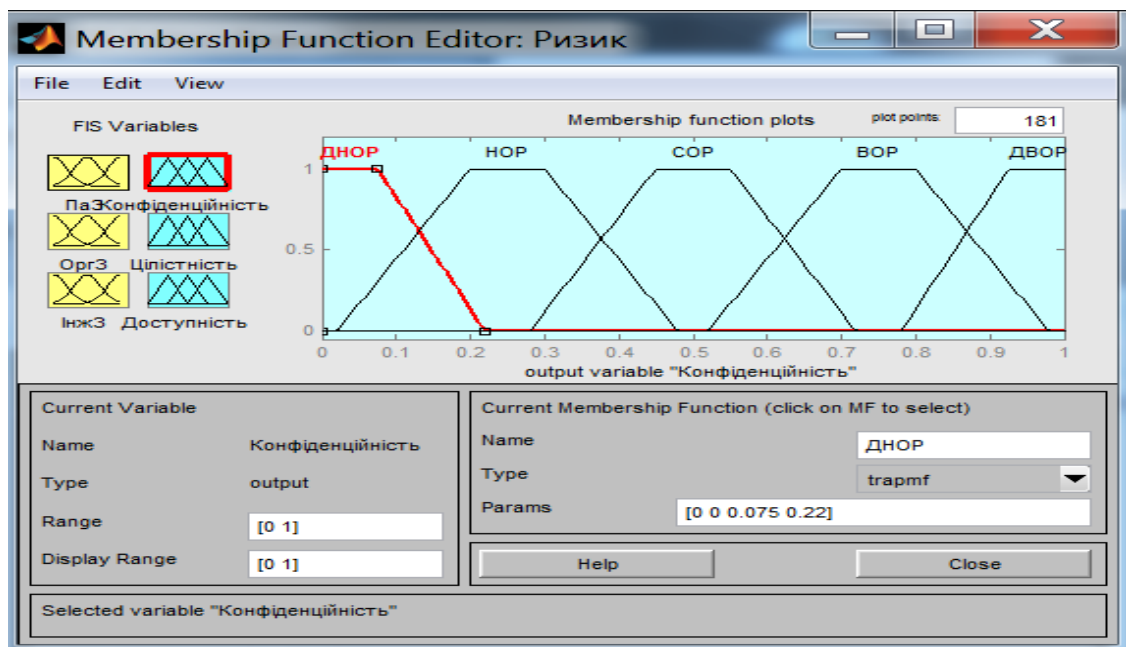


Рис.3.88. Вихідна змінна "Конфіденційність"

Де $T = \{ \text{Дуже низька очевидність ризику (ДНОР)}; \text{Низька очевидність ризику (НОР)}; \text{Середня очевидність ризику (СОР)}; \text{Висока очевидність ризику (ВОР)}; \text{Дуже висока очевидність ризику (ДВОР)} \}$

Висока очевидність ризику (ВОР); Дуже висока очевидність ризику (ДВОР)}.

ДНОР – [0 0 0,075 0,22];

НОР – [0,02 0,2 0,3 0,48];

СОР – [0,28 0,45 0,55 0,72];

ВОР – [0,52 0,7 0,8 0,98];

ДВОР – [0,78 0,925 1 1].

Далі необхідно визначити правила нечіткого виведення для методики оцінки та аналізу ризиків (експертної системи). Ці правила є алгоритм, оцінки ризиків.

Щоб зрозуміти, яким чином рівні програмно-апаратного, організаційно-правового та інженерно-технічного захисту впливають на вихідні змінні – порушення конфіденційності, цілісності, доступності, складаємо матрицю (табл. 3.5).

Таблиця 3.5. Рівні впливу на виникнення ризиків

<i>Рівень/Ризик</i>	К	Ц	Д
<i>Паз</i>	3	1	2
<i>ОргЗ</i>	1	2	3
<i>ІнжЗ</i>	2	3	1

У матриці визначено рівні впливу на виникнення ризиків. Наприклад, на конфіденційність інформації в організації найбільший вплив має рівень програмно-апаратного захисту, тому на перетині першого рядка та першого стовпчика матриці виставляється значення 3 (високий). Далі всі інші рівні ранжують за тим же принципом. Кожен з термів буде відповідати значенню з матриці:

1 – низький;

2 – середній;

3 – високий.

Для кращого сприйняття, наведемо приклад. Якщо рівень ПАЗ = низький (н), ОргЗ = середній (с), а рівень ІнжЗ = високий (в), то ризик порушення конфіденційності (к) - середній, цілісності (ц) – низький, доступності (д) – середній. Це було розраховано таким чином: з матриці випливає, що на конфіденційність найбільший вплив має рівень ПАЗ, потім – ІнжЗ, тому що рівень ПАЗ у даному прикладі низький, а рівень ІнжЗ – високий, то ризик порушення конфіденційності середній $(н+в) = с$, для інших ризиків розрахунок відбувається аналогічним чином. Необхідно також ввести правило: $(с + в) = с$, $(в + с) = в$.

База правил нечіткого виводу, побудована на основі раніше наведеної матриці, наведена на рис. 3.89.

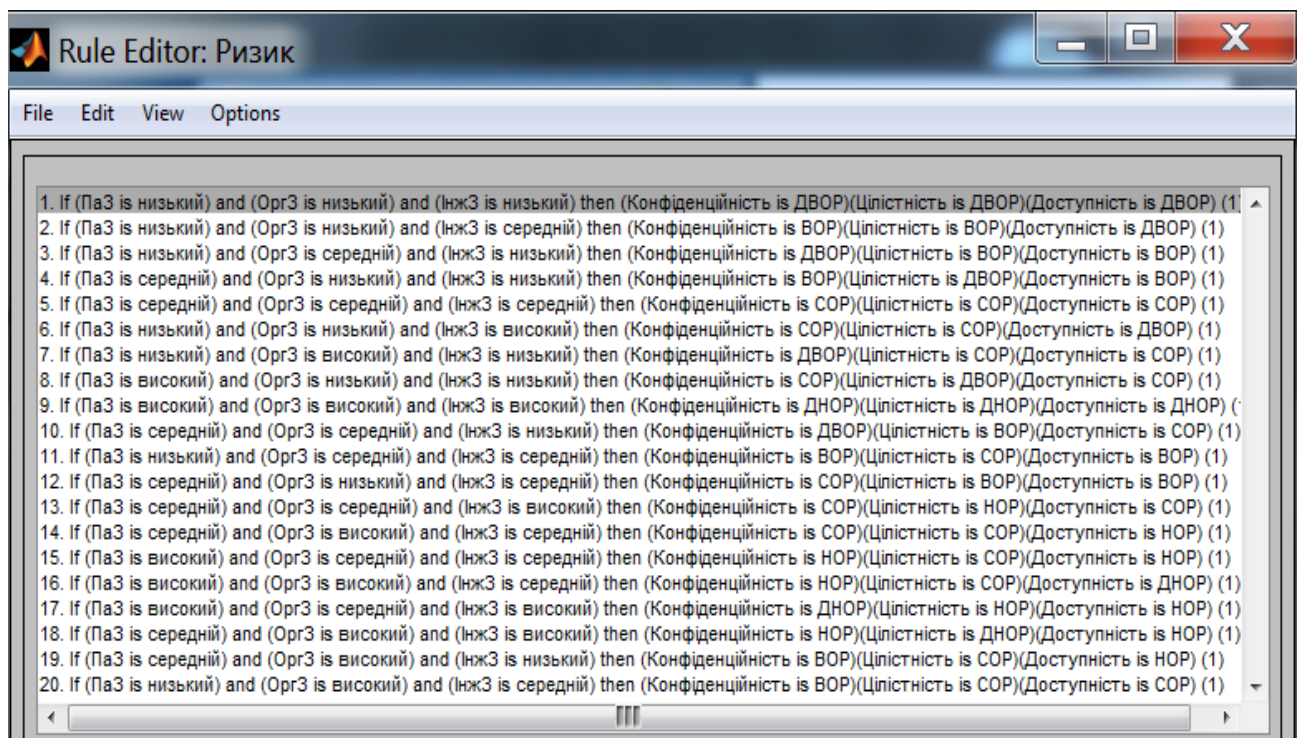


Рис. 3.89. Правила нечіткого виводу

Припустимо, що на основі експертних даних були отримані оцінки програмно-апаратного, інженерно-технічного та організаційного захисту, які будемо вводити у вікно механізму виведення графічного інтерфейсу Fuzzy Logic Toolbox (рис. 3.90).

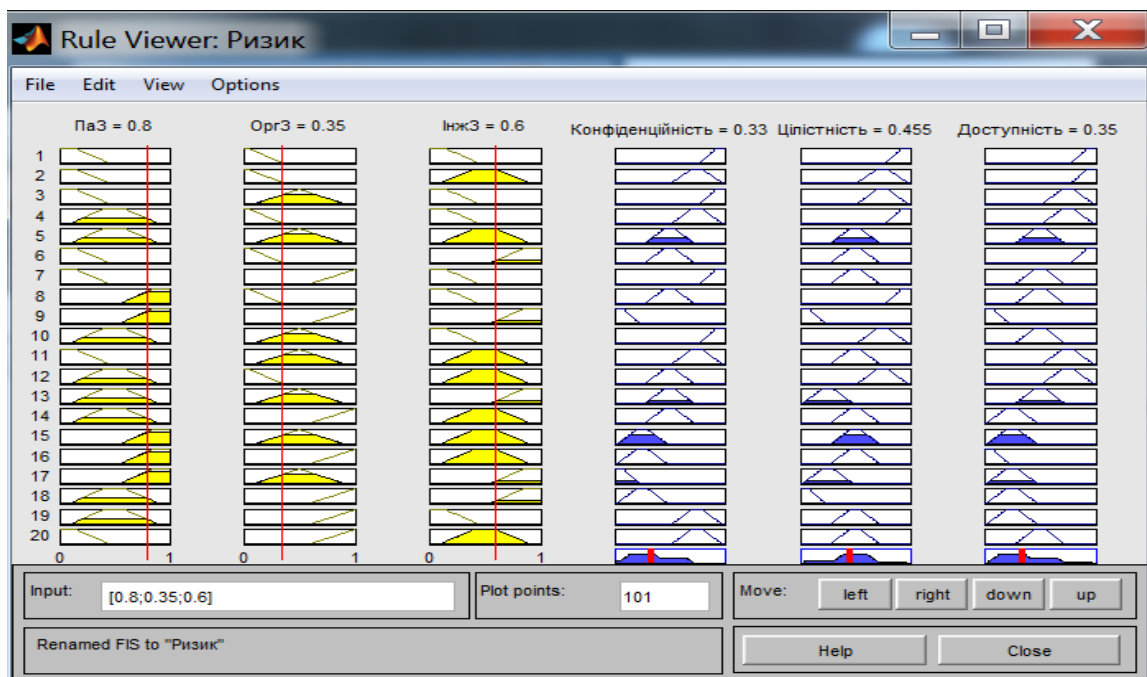


Рис. 3.90. Вікно виводу

ПАЗ = 0,8, що відповідає терму "високий рівень забезпечення І";

ОргЗ = 0,35, що відповідає терму "задовільна для забезпечення початкового рівня захисту";

ІнжЗ = 0,6, що відповідає терму "достатня для базової інформаційного захисту".

З рисунку 3.90 видно, що при уведених значеннях вхідних змінних, вихідні змінні приймають значення: конфіденційність = 0,33, що відповідає терму "низька очевидність ризику"; цілісність = 0,455, що відповідає терму "середня очевидність ризику"; доступність = 0,35, що відповідає терму "низька очевидність ризику".

Графічний інтерфейс програмного інструментарію дозволяє отримати графік залежності вихідної величини від кожної з вхідних змінних.

На рис. 3.91 представлений графік залежності вихідної змінної "Цілісність" від вхідної змінної ОргЗ.

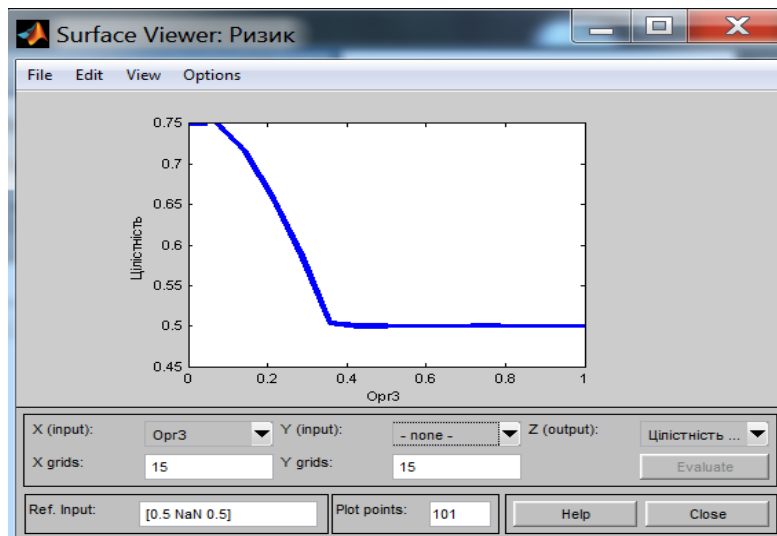


Рис.3.91. Залежність змінної "Цілісність" від ОргЗ

Графік показує зворотну залежність величини ризику порушення цілісності від рівня організаційної захисту.

На рис. 3.92 наведена отримана поверхня залежності вихідної лінгвістичної змінної від двох вхідних з фіксованим значенням третин змінної для бази правил нечіткої моделі.

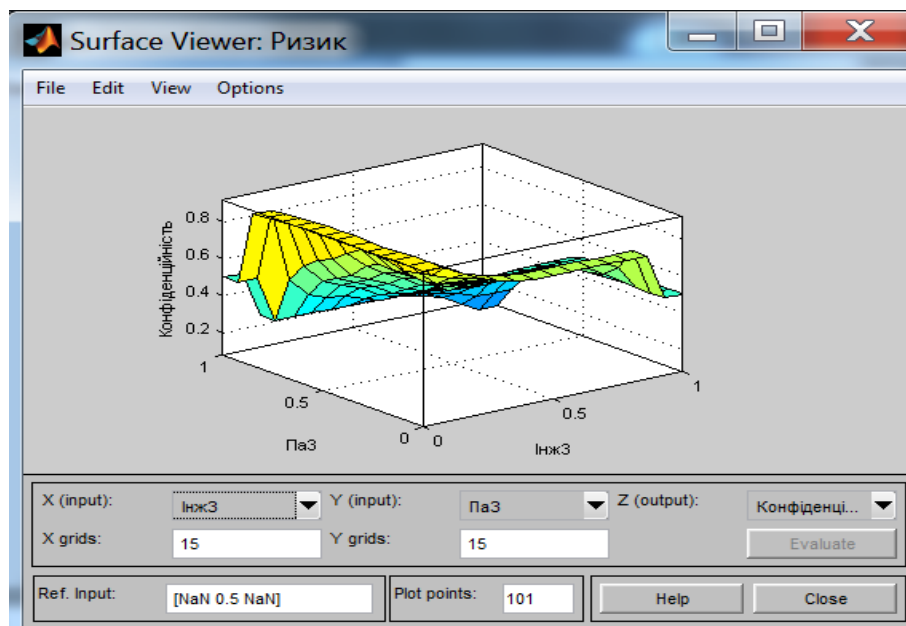


Рис. 3.92. Поверхня системи нечіткої моделі

Висновки. Дана методика дає можливість оцінювати ризики інформаційної безпеки з використанням нечіткої логіки на базі інструментарію MATLAB і дозволяє наочно уявити стан системи захисту інформації, а також комплексно оцінити можливі загрози безпеки й отримати оцінки інформаційних ризиків.

Методика може використовуватися в організаціях малого і середнього бізнесу для базового аналізу ризиків інформаційної безпеки і послужити для оцінки ефективності системи захисту інформації в організації.

Створена в ході розробки методики база правил може бути змінена залежно від пріоритетів кожного з рівнів захисту. Також для її побудови можуть використовуватися різні методи прийняття рішень.

Розроблена нечітка продукційна модель дозволяє істотно розширити можливості існуючих методик, зняти обмеження на кількість урахуваних вхідних змінних та інтегрувати як якісні, так і кількісні підходи до оцінки ризиків. Використовувані в методиці механізми оцінки ризику на основі нечіткої логіки дозволяють отримати лінгвістичний опис ступеня ризику, що дозволяє ІТ-менеджерам виявити пріоритети ризиків (дуже низька очевидність ризику; низька очевидність ризику; середня очевидність ризику; висока очевидність ризику; дуже висока очевидність ризику) і вибрати план заходів щодо зниження рівня найбільш небезпечних загроз інформаційній безпеці організації.

Основна складність механізму отримання оцінок ризику на основі нечіткої логіки полягає в побудові моделі для проведення лінгвістичного аналізу ризиків системи забезпечення інформаційної безпеки, однак даний механізм є ефективним інструментом, коли інші підходи до оцінки ризику неприйнятні.

Він володіє широкими можливостями і дозволяє адаптувати його до наявних на підприємстві моделей управління ризиками, а також модифікувати з урахуванням реальних умов політики інформаційної безпеки організації.

3.10. Нечіткі нейронні мережі

Нечіткі нейронні мережі або гібридні мережі покликані об'єднати в собі переваги нейронних мереж і систем нечіткого виводу. З одного боку, вони дозволяють розробляти і представляти моделі систем у формі правил нечітких продукцій, які володіють наочністю та простотою змістовної інтерпретації. З іншого боку, для побудови правил нечітких продукцій використовуються методи нейронних мереж, що є більш зручним і менш трудомістким процесом для системних аналітиків. Останнім часом апарат гібридних мереж повсюдно визнається фахівцями як один з найбільш перспективних для вирішення слабо або погано структурованих задач прикладного системного аналізу.

Нижче розглянуті особливості побудови та використання гібридних мереж, реалізовані в пакеті Fuzzy Logic Toolbox. Як вже говорилося вище, у даний час запропоновані різні схеми класифікації нейронних мереж і відповідні алгоритми їх навчання. Одним з найпоширеніших алгоритмів навчання є так званий алгоритм зворотного поширення помилки (backpropagation). Цей алгоритм являє собою ітеративний градієнтний алгоритм мінімізації середньоквадратичного відхилення значень виходу від бажаних значень (мінімізації помилки) у багатошарових нейронних мережах.

Нечітка нейронна мережа – це багатошарова НМ, в якій шари виконують функції елементів системи нечіткого виводу. Характерною особливістю цих мереж є можливість використання нечітких правил виводу для розрахунку вихідного сигналу. Нейрони цієї мережі характеризуються набором параметрів, налаштування яких відбувається в процесі навчання, як у звичайних НМ. Наприклад, для нечіткої нейронної мережі на базі алгоритму Сугено 1-й шар здійснює фазифікацію, нелінійні функції $\mu_{r,j}(x_j)$, де r – номер продукційного правила, j – номер компонента вхідного вектора x відповідають функцій

приналежності передумов правил. Налаштовані параметри даного шару – параметри функцій приналежності. Шар 2 здійснює обчислення результируючих функцій приналежності передумов нечітких правил. Шар 3, що складається з двох нейронів-суматорів, здійснює підсумовування і зважене підсумовування вихідних сигналів шару 2. Параметрами цього шару є вагові коефіцієнти w_r . Шар 4, що складається з єдиного вихідного нейрона, реалізує операцію ділення $z = f_1 / f_2$ і не містить параметрів, що настроюються. Навчання нечітких мереж, так само як і класичних мереж, може проводитися або за алгоритмом з учителем, заснованим на мінімізації цільової функції, або за алгоритмом самоорганізації [11,21].

Вибір виду і структури нейронної мережі зумовлюється специфікою розв'язуваної задачі. При цьому для вирішення окремих типів практичних завдань розроблені оптимальні конфігурації нейронних мереж, які найбільш адекватно відображають особливості відповідної проблемної області. Подальшим розвитком нейронних мереж є так звані гібридні мережі.

Гібридна мережа як адаптивна система нейронечіткого виводу

Гібридна мережа являє собою багатозарову нейронну мережу спеціальної структури без зворотних зв'язків, у якій використовуються звичайні (не нечіткі) сигнали, ваги і функції

активації, а виконання операції підсумовування $s = \sum_{i=1}^n w_i x_i + b_i$

засноване на використанні фіксованого Т-норми, Т-конорми або деякої іншої безперервної операції. При цьому значення входів, виходів і ваг гібридної нейронної мережі є речовими числами з відрізка $[0,1]$.

Основна ідея, покладена в основу моделі гібридних мереж, полягає в тому, щоб використовувати існуючу вибірку даних для визначення параметрів функцій приналежності, які найкраще відповідають деякій системі нечіткого виводу. При цьому для

знаходження параметрів функцій приналежності використовуються відомі процедури навчання нейронних мереж.

У пакеті Fuzzy Logic Toolbox системи MATLAB гібридні мережі реалізовані у формі так званої адаптивної системи нейронечіткого виведення ANFIS. З одного боку, гібридна мережа ANFIS являє собою нейронну мережу з єдиним виходом і кількома входами, які являють собою нечіткі лінгвістичні змінні. При цьому терми вхідних лінгвістичних змінних описуються стандартними для системи MATLAB функціями приналежності, а терми вихідної змінної представляються лінійної або постійною функцією приналежності. З іншого боку, гібридна мережа ANFIS являє собою систему нечіткого виводу FIS типу Сугено нульового або першого порядку, в якій кожне з правил нечітких продукцій має постійну вагу, рівну 1. В системі MATLAB користувач має можливість редагувати і налаштовувати гібридні мережі ANFIS аналогічно системі нечіткого виводу, використовуючи всі розглянуті раніше засоби пакету FuzzyLogicToolbox.

Реалізація ANFIS в середовищі MATLAB. У пакеті Fuzzy Logic Toolbox гібридні мережі реалізовані у формі адаптивних систем нейро-нечіткого виводу ANFIS. При цьому розробка та дослідження гібридних мереж виявляється можливою:

- в інтерактивному режимі з допомогою спеціального графічного редактора адаптивних мереж, що отримав назву редактора ANFIS;
- в режимі командного рядка за допомогою введення імен відповідних функцій з необхідними аргументами безпосередньо у вікно команд системи MATLAB. Для роботи в режимі командного рядка призначені спеціальні функції.

Реалізація ANFIS в середовищі MATLAB

У пакеті Fuzzy Logic Toolbox гібридні мережі реалізовані у формі адаптивних систем нейро-нечіткого виводу ANFIS. При цьому розробка та дослідження гібридних мереж виявляється можливою:

- в інтерактивному режимі з допомогою спеціального графічного редактора адаптивних мереж, що отримав назву редактора ANFIS;

- у режимі командного рядка за допомогою введення імен відповідних функцій з необхідними аргументами безпосередньо у вікно команд системи MATLAB. Для роботи в режимі командного рядка призначені спеціальні функції.

Редактор ANFIS дозволяє створювати або завантажувати конкретну модель адаптивної системи нейронечіткого виводу, виконувати її навчання, візуалізувати її структуру, змінювати і налаштовувати її параметри, а також використовувати налагоджену мережу для отримання результатів нечіткого виводу. Графічний інтерфейс редактора ANFIS викликається функцією `anfisedit` з командного рядка (рис.3.93).

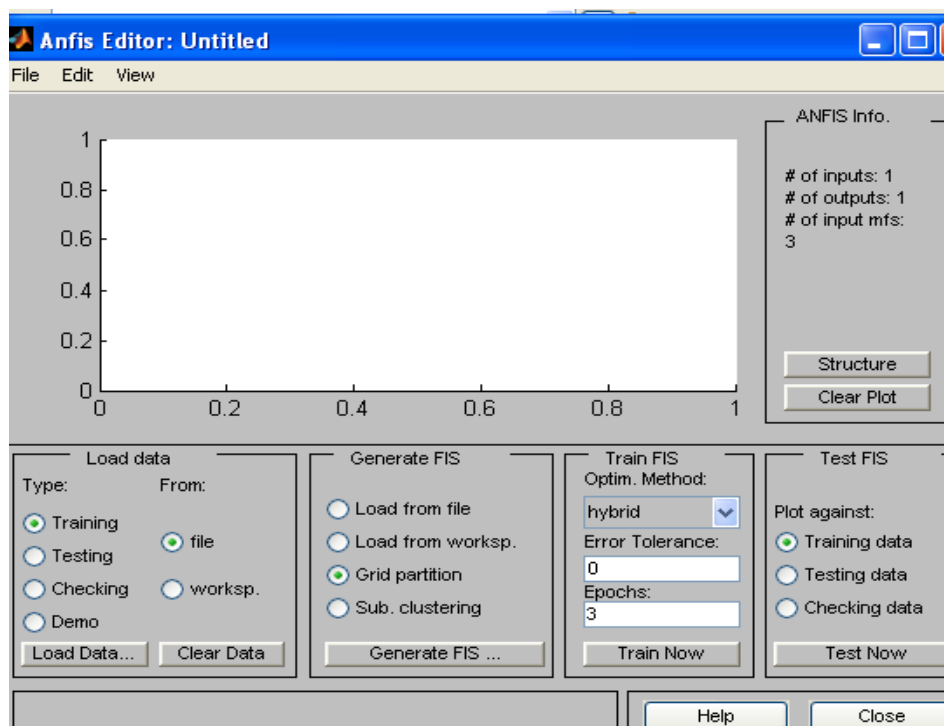


Рис.3.93. Вікно редактора гібридних систем

За допомогою даного редактора здійснюється створення або завантаження структури гібридної системи, перегляд структури, налаштування її параметрів, перевірка якості функціонування такої системи.

Основну частину графічного інтерфейсу займає вікно візуалізації даних, яке розташоване нижче головного меню. Для новоствореної гібридної мережі це вікно не містить ніяких даних.

Для створення гібридної мережі необхідно завантажити дані. Для цієї мети слід скористатися кнопкою Load Data у лівій нижній частині графічного вікна. При цьому дані можуть бути завантажені із зовнішнього файлу (disk) або робочої області (worksp.). У першому випадку необхідно попередньо створити файл з вихідними даними (файл з розширенням .dat), який являє собою звичайний текстовий файл. При цьому вихідні дані являють собою звичайну числову матрицю розмірності $m \times (n + 1)$, в якій кількість рядків m відповідає обсягу вибірки, перші n стовпців значень вхідних змінних моделі, а останній стовпець значенню вихідної змінної. Згідно з правилами системи MATLAB окремі значення матриці відокремлюються пробілами, а кожен рядок матриці завершується символом "переклад каретки" (клавіша <Enter>).

Завантажувані вихідні дані можуть бути одного з таких типів:

- навчальні дані (Training) – обов'язкові дані, які використовуються для побудови гібридної мережі;
- тестові дані (Testing) – необов'язкові дані, які використовуються для тестування побудованої гібридної мережі з метою перевірки якості функціонування побудованої гібридної мережі;
- перевірочні дані (Checking) – необов'язкові дані, які використовуються для перевірки побудованої гібридної мережі з метою з'ясування факту перенавчання мережі;
- демонстраційні дані (Demo) – дозволяють завантажити один з демонстраційних прикладів гібридної мережі.

Після завантаження навчальних даних їх структура буде відображена у робочому вікні редактора ANFIS. При цьому кожному рядку даних відповідає окрема точка графіка, яка для

навчальних даних зображується колом. На горизонтальній осі зазначаються порядковий номер (індекс) окремого рядка даних, а вертикальна вісь служить для вказівки значень вихідної змінної. Після підготовки і завантаження навчальних даних можна створити структуру системи нечіткого виводу FIS типу Сугено, яка є моделлю гібридної мережі. Для цієї мети слід скористатися кнопкою Generate FIS у нижній частині робочого вікна редактора. При цьому дві перші опції відносяться до попередньо створеній структурі гібридної мережі, а дві останніх до форми розбиття вхідних змінних моделі.

Завантажити структуру вже створеної FIS можна або з диска (Load from disk), або з робочої області (Load from worksp.). При створенні нової структури FIS можна незалежно розбити всі вхідні змінні області їх значень (Grid partition) або скористатися процедурою субтрактивної кластеризації для попереднього розбиття значень вхідних змінних на кластери близьких значень (Sub. clustering).

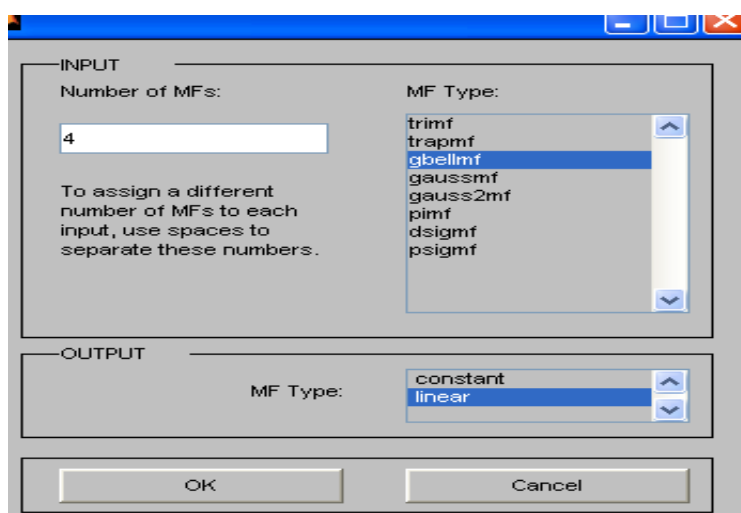


Рис.3.94. Діалогове вікно для завдання кількості і типу функцій приналежності

Після натискання кнопки Generate FIS викликається діалогове вікно із зазначенням кількості та типу функцій приналежності для окремих термів вхідних змінних і вихідної

змінної (рис.3.94). У цьому випадку можна вибрати будь-який тип функцій належності з реалізованих у системі MATLAB.

Після генерації структури гібридної мережі можна візуалізувати її структуру, для чого слід натиснути кнопку Structure у правій частині графічного вікна. Структура отриманої в результаті системи нечіткого виводу FIS відображається в окремому вікні (рис. 3.95).

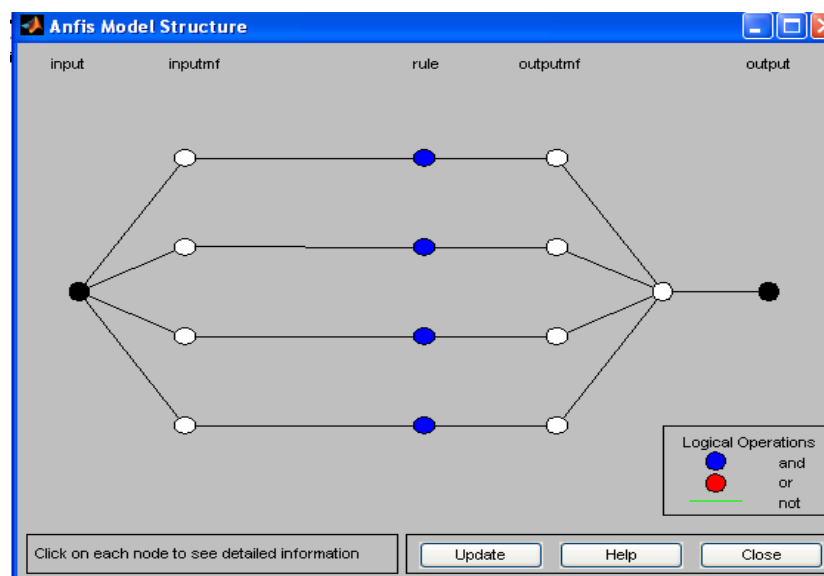


Рис.3.95. Структура згенерованої системи нечіткого виводу

Перед навчанням гібридної мережі необхідно задати параметри навчання, для чого слід скористатися такою групою опцій у правій нижній частині робочого вікна:

1. Вибрати метод навчання гібридної мережі зворотного поширення (backprop) або гібридний (hybrid), що представляє собою комбінацію методу найменших квадратів та методу зменшення зворотного градієнта.

2. Встановити рівень помилки навчання (Error Tolerance) за замовчуванням значення 0 (змінювати не рекомендується).

3. Задати кількість циклів навчання (Epochs) за замовчуванням значення 3 (рекомендується збільшити і для розглянутого прикладу задати його значення рівним 40).

Для навчання мережі слід натиснути кнопку Gain Now. При цьому хід процесу навчання ілюструється у вікні візуалізації у формі графіка залежності помилки від кількості циклів навчання. У цьому випадку на верхньому графіку зображено залежність помилки перевірки від кількості циклів навчання, а на нижньому графіку залежність помилки навчання від кількості циклів навчання (знаком "*").

Аналогічно можуть бути виконані додаткові етапи тестування та перевірки гібридної мережі, для яких необхідно попередньо завантажити відповідні дані.

3.10.1. Приклади розв'язання задачі нейро-нечіткого висновку

Для ілюстрації процесу розробки гібридної мережі у системі MATLAB розглянемо задачу побудови адаптивної системи нейронечіткого висновку для апроксимації деякої залежності, яка описується математичною функцією $y = x^3$. Цей приклад дозволяє не тільки уточнити зміст і послідовність етапів розробки, але і оцінити точність отриманої нечіткої моделі шляхом порівняння прогнозованих модельних значень із заздалегідь відомими значеннями відповідної функції.

Загальна послідовність процесу розробки моделі гібридної мережі може бути представлена у такому вигляді.

1. Для початку за допомогою редактора-налагоджувача m-файлів підготуємо навчальні дані, які містять 9 рядків пар "значення вхідної змінної – значення вихідної змінної" такого вигляду (рис. 3.96). Збережемо навчальні дані у зовнішньому файлі з іменем function1.dat.

2. Далі завантажимо цей файл з навчальними даними у редактор ANFIS (рис. 3.97). У робочому вікні редактора буде зображено графік, форма якого аналогічна вихідної математичної функції.

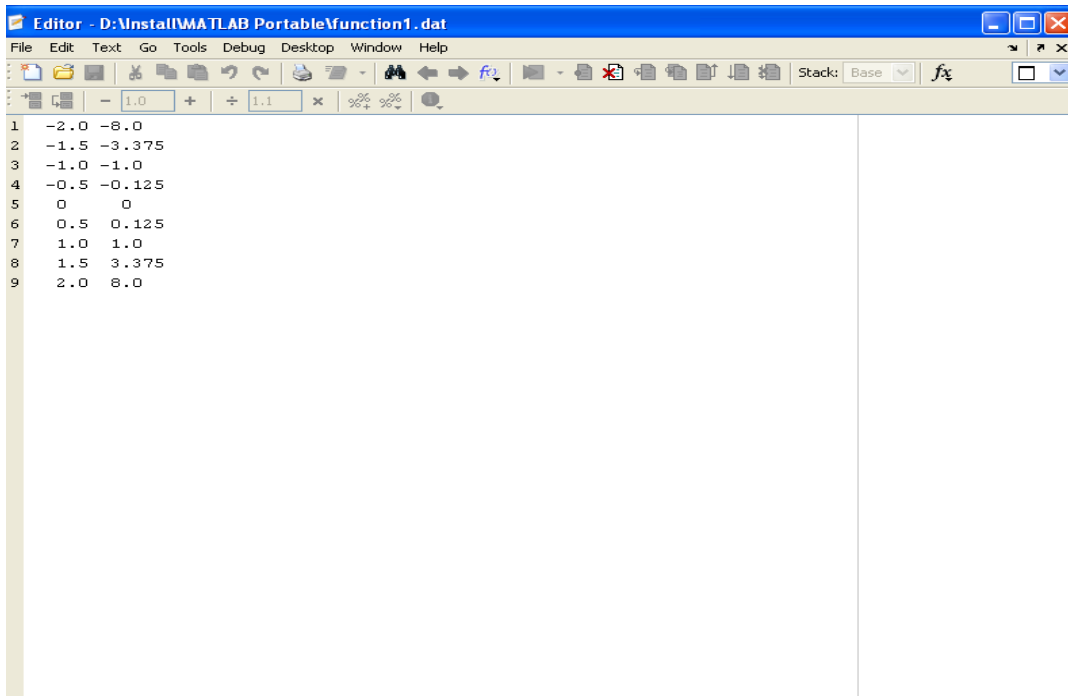


Рис.3.96. Навчальні дані для прикладу побудови гібридної мережі ANFIS представляє функцію $y = x^3$

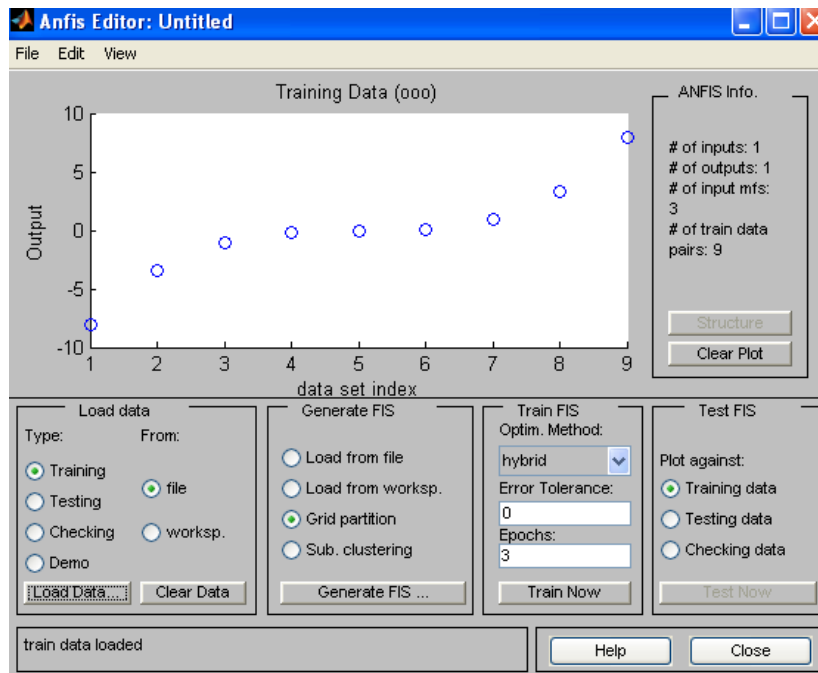


Рис.3.97. Графічний інтерфейс редактора ANFIS після завантаження файлу function.dat з навчальними даними

3. Оскільки в даному випадку відсутні тестові та перевіірочні дані, можна відразу приступити до генерації структури системи нечіткого висновку FIS. Встановивши параметри генерації, отримаємо структуру FIS, вигляд якої також збігається із зображеною на рис. 3.95.

4. Тепер можна перейти до навчання згенерованої системи нечіткого висновку. Для цього залишимо без зміни запропоновані системою MATLAB за замовчуванням метод навчання (гібридний) і рівень помилки (0), а кількість циклів навчання змінимо на 40. Після навчання мережі в робочому вікні редактора ANFIS буде зображено графік зміни помилки в ході виконання окремих циклів навчання (рис.3.98).

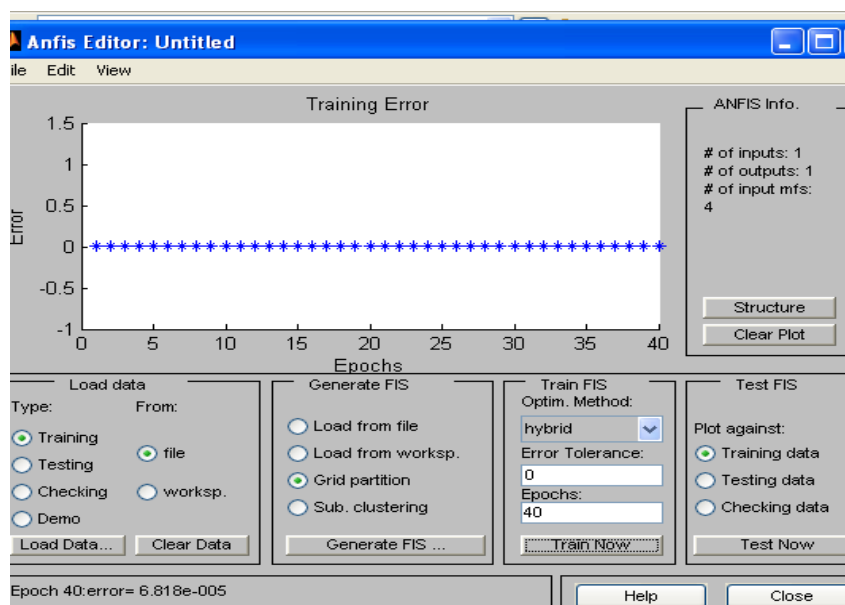


Рис.3.98.Графік залежності помилки навчання від кількості циклів навчання

5. Виконати аналіз точності побудованої нечіткої моделі гібридної мережі можна за допомогою перегляду поверхні відповідної системи нечіткого висновку (рис. 3.99).

Візуальний аналіз зображеного графіка з точним графіком функції $y = x^3$ дозволяє судити про досить високий ступень їх збігу, допоможе свідчити про адекватність побудованої нечіткої моделі гібридної мережі.

Аналіз адекватності побудованої моделі можна виконати за допомогою перегляду правил відповідної системи нечіткого висновку (рис. 3.100).

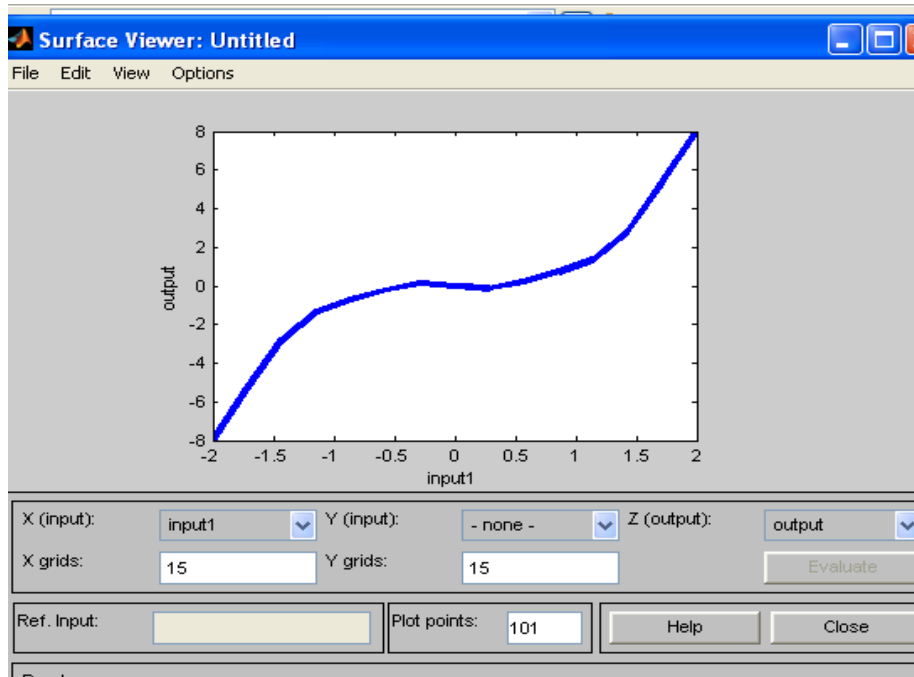


Рис.3.99. Графічний інтерфейс перегляду поверхні згенерованої системи нечіткого висновку

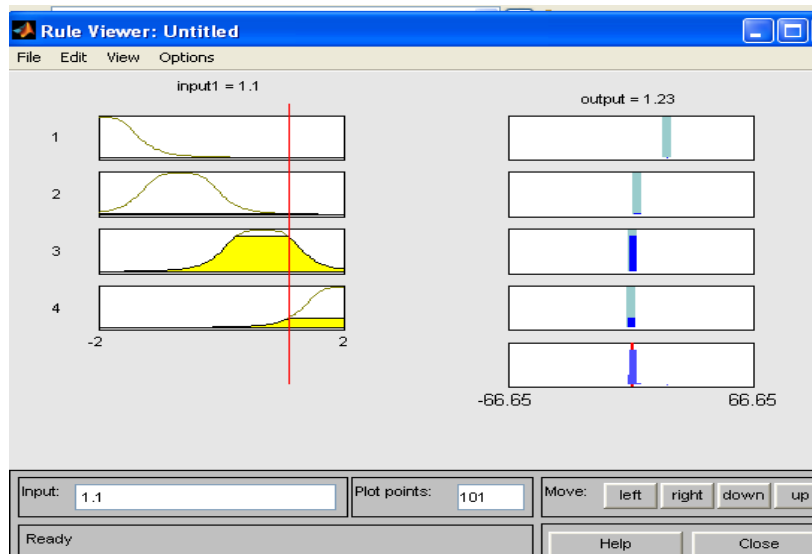


Рис.3.100. Графічний інтерфейс перегляду правил згенерованої системи нечіткого висновку

Перевірка побудованої моделі гібридної мережі може бути виконана для декількох значень вихідної змінної. З цією метою необхідно ввести конкретне значення в поле вводу Input (наприклад, значення 1.1), після натискання клавіші <Enter> за допомогою побудованої моделі буде отримано відповідне значення вихідної змінної (у даному випадку значення 1.23). Порівнюючи отримане значення з точним значенням функції 1.331), отримуємо відносну помилку близько 8%.

Менш вдалою виявляється перевірка для значення вхідної змінної 0.1, для якого побудована модель пропонує від'ємне значення $-0,148$. Очевидно, цей факт свідчить не на користь адекватності побудованої нечіткої моделі і вимагає додаткового налаштування.

У загальному випадку додаткове налаштування моделі може бути виконана декількома способами. Найбільш прийнятними з них представляються такі.

1. Підготовка і завантаження більшого за обсягом вибірки файлу з навчальними вихідними даними.

2. Підготовка і завантаження додаткового файлу з перевірочними вихідними даними, сформованими для пар значень розглянутої математичної функції, відсутніх у вибірці навчальних даних.

3. Редагування типів і параметрів функцій належності термів вхідної і вихідної змінних за допомогою редактора функцій приналежності системи MATLAB.

Проілюструємо третій спосіб додаткового налаштування побудованої нечіткої моделі гібридної мережі. На перший погляд він видається найбільш природним з точки зору можливості візуального контролю виконуваних змін параметрів. З цією метою відкриємо редактор функцій приналежності та методом підбору змінимо кількісні значення параметрів другої і третьої функції належності вхідної змінної, оскільки саме вони "працюють" при отриманні некоректного значення вихідної змінної: 0.148 для значення вхідної змінної 0.2. (рис.3.101).

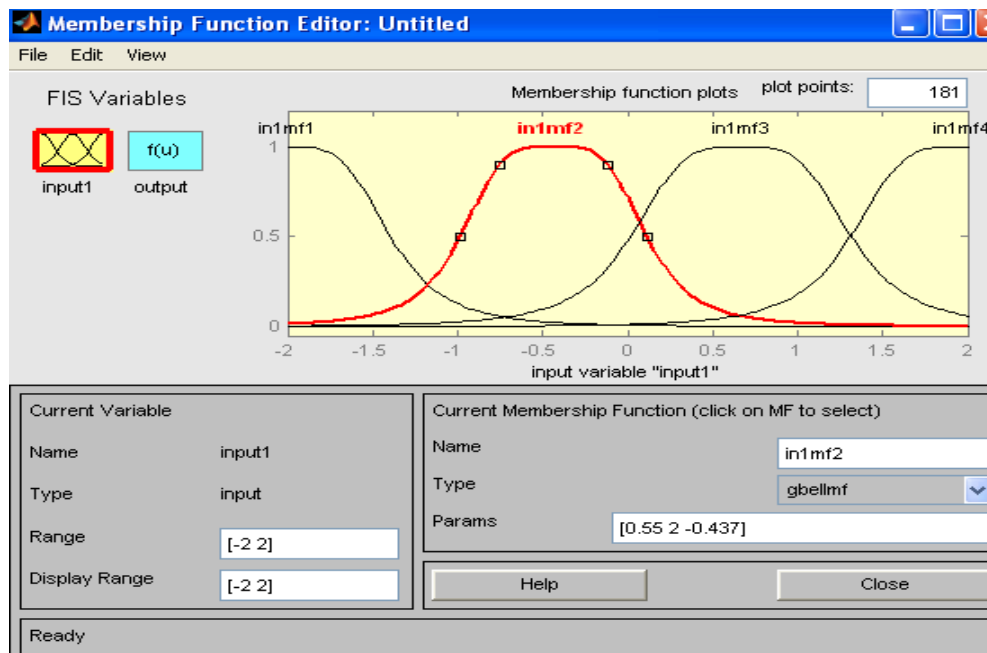


Рис.3.101. Графічний інтерфейс редактора функцій приналежності побудованої системи нечіткого висновку

Після зміни параметрів у полі Params для другої функції належності на значення $[0.55 \ 2 \ 0.437]$ і для третьої функції належності на значення $[0.65 \ 1.9 \ 0.667]$ одержимо більш точне значення вихідної змінної для вихідного значення вхідної змінної 0.2 (рис.3.102).

Більш ефективним способом налаштування, а точніше модифікації даної нечіткої моделі виявився перший. З цією метою необхідно збільшити обсяг навчальної вибірки до 17 пар значень.

На закінчення слід зазначити, що навіть найпростіші розглянуті приклади відображають творчий характер процесу побудови і аналізу моделей гібридних мереж. При цьому вибір того чи іншого способу додаткового налаштування нечітких моделей залежить не тільки від специфіки розв'язуваної задачі, але і від обсягу доступної вибірки навчальних та перевірочних даних. У разі недостатньої інформації навчальних даних використання гібридних мереж може виявитися взагалі недоцільним, оскільки отримати адекватну нечітку модель, а значить і точний прогноз значень вихідної змінної не представляється можливим.

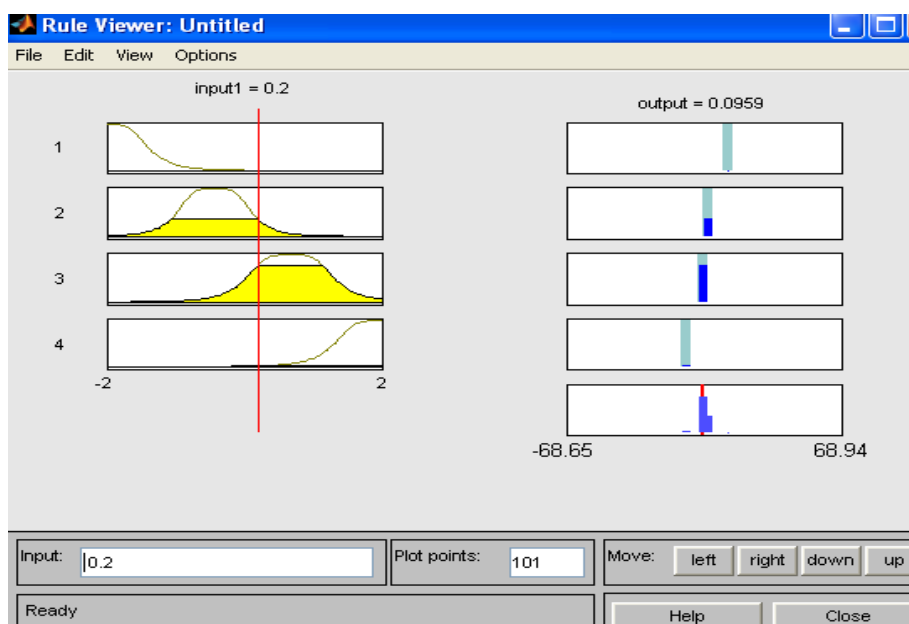


Рис.3.102. Результат ручного налаштування параметрів функцій належності входньої змінної для розглянутої системи нечіткого висновку

Саме з цих причин необхідний попередній аналіз всіх можливостей застосовуваних нечітких моделей для вирішення конкретних завдань у тій чи іншій проблемній області. Такий аналіз необхідно виконувати з системної точки зору і з урахуванням всіх обставин, що складаються на даний момент. Тільки всебічна і повна оцінка проблемної ситуації дозволить розробити адекватну модель вирішення тієї чи іншої конкретної задачі нечіткого управління або прийняття рішень.

Аналіз і прогнозування валютних цін на фінансовому ринку

В якості другого прикладу побудови і використання адаптивної системи нейронечіткого висновку розглянемо процес розробки нечіткої моделі гібридної мережі для вирішення задачі прогнозування валютних цін на фінансовому ринку. Суть цього завдання полягає в тому, щоб, знаючи динаміку зміни курсової вартості продажу певної валюти за фіксований інтервал часу, передбачити її значення курсової вартості на певний момент часу в майбутньому. При цьому характерною особли-

вістю динаміки зміни курсу (тренду) є наявність двох основних тенденцій в коливаннях відповідних цін.

З одного боку, спостерігається загальне довгострокове підвищення курсової вартості, пов'язане з величиною інфляції. З іншого боку, спостерігається короткострокове коливання цін, пов'язане з цілим рядом випадкових факторів, адекватне представлення яких у тій чи іншій формальній моделі навряд чи можливе.

Традиційно для вирішення цього завдання застосовуються різні моделі технічного аналізу, засновані на використанні різних індикаторів. У той же час наявність неявних тенденцій у динаміці зміни курсової вартості валют дозволяє застосувати модель адаптивних нейронечітких мереж.

В якості вихідних даних можна скористатися інформацією про динаміку курсу ЦБ РФ по валюті Долар США (USD) за певний часовий інтервал. Візьмемо значення курсової вартості USD за 1 одиницю в період з 1 жовтня 2002 р. по 10 грудня 2002 р. Дану інформацію для зручності подальшої роботи представимо в табличній формі (табл.3.6).

Припустимо, що нечітка модель гібридної мережі буде містити 4 вхідних змінних. При цьому перша вхідна змінна відповідатиме курсу USD на поточний банківський день, друга курсу USD на попередній банківський день, тобто на день $(i-1)$, де i позначений поточний банківський день. Тоді третя вхідна змінна відповідатиме курсу USD на $(i-2)$ банківський день, а четверта курсу USD на $(i-3)$ банківський день.

Відповідні навчальні дані можуть бути зведені в окрему таблицю. Обсяг отриманої таким чином навчальної вибірки дорівнює 40 (табл. 3.7), що відповідає динаміці курсу USD у період з 4 жовтня 2002 року по 29 листопада 2002 року.

При цьому дані за грудень 2002 року не увійшли до складу навчальної вибірки і можуть бути використані для перевірки адекватності побудованої нечіткої моделі.

Таблиця 3.6. Динаміка курсу іноземної валюти (USD) у період з 1.10.02 за 10.12.02

Дата	Курс USD	Дата	Курс USD	Дата	Курс USD
01.10.02	31.6827	01.11.02	31.7701	03.12.02	31.8547
02.10.02	31.6919	02.11.02	31.7646	04.12.02	31.8584
03.10.02	31.6982	05.11.02	31.7744	05.12.02	31.8596
04.10.02	31.6809	06.11.02	31.7909	06.12.02	31.8578
05.10.02	31.68	07.11.02	31.7756	07.12.02	31.86
08.10.02	31.6795	11.11.02	31.7756	10.12.02	31.8597
09.10.02	31.6799	12.11.02	31.7756		
10.10.02	31.6803	13.11.02	31.8226		
11.10.02	31.6685	14.11.02	31.8157		
12.10.02	31.6703	15.11.02	31.8203		
15.10.02	31.6703	16.11.02	31.8225		
16.10.02	31.6762	19.11.02	31.8231		
17.10.02	31.6767	20.11.02	31.8224		
18.10.02	31.6761	21.11.02	31.823		
19.10.02	31.6727	22.11.02	31.8248		
22.10.02	31.6973	23.11.02	31.8222		
23.10.02	31.7272	26.11.02	31.8416		
24.10.02	31.7159	27.11.02	31.8382		
25.10.02	31.7109	28.11.02	31.84		
26.10.02	31.7314	29.11.02	31.84		
29.10.02	31.7411	30.11.02	31.8424		
30.10.02	31.6977				
31.10.02	31.7408				

Збережемо навчальну вибірку у зовнішньому файлі під ім'ям priceUSD.dat. Після цього відкриємо редактор ANFIS, у який завантажимо цей файл з навчальними даними. Зовнішній вигляд редактора ANFIS із завантаженими навчальними даними зображено на рис. 3.103. Перед генерацією структури системи нечіткого висновку типу Сугено після виклику діалогового вікна властивостей задамо для кожної з вхідних змінних на 3 лінгвістичних терми, а в якості типу їх функцій приналежності виберемо трикутні функції (установлені системою MATLAB за замовчуванням).

Таблиця 3.7. Навчальні дані для побудови моделі гібридної мережі

Перша вхідна змінна	Друга вхідна змінна	Третя вхідна змінна	Четверта вхідна змінна	Вихідна змінна
31.6809	31.6982	31.6919	31.6827	31.68
31.68	31.6809	31.6982	31.6919	31.6795
31.6795	31.68	31.6809	31.6982	31.6799
31.6799	31.6795	31.68	31.6809	31.6803
31.6803	31.6799	31.6795	31.68	31.6685
31.6685	31.6803	31.6799	31.6795	31.6703
31.6703	31.6685	31.6803	31.6799	31.6703
31.6703	31.6703	31.6685	31.6803	31.6762
31.6762	31.6703	31.6703	31.6685	31.6767
31.6767	31.6762	31.6703	31.6703	31.6761
31.6761	31.6767	31.6762	31.6703	31.6727
31.6727	31.6761	31.6767	31.6762	31.6973
31.6973	31.6727	31.6761	31.6767	31.7272
31.7272	31.6973	31.6727	31.6761	31.7159
31.7159	31.7272	31.6973	31.6727	31.7109
31.7109	31.7159	31.7272	31.6973	31.7314
31.7314	31.7109	31.7159	31.7272	31.7411
31.7411	31.7314	31.7109	31.7159	31.6977
31.6977	31.7411	31.7314	31.7109	31.7408
31.7408	31.6977	31.7411	31.7314	31.7701
31.7701	31.7408	31.6977	31.7411	31.7646
31.7646	31.7701	31.7408	31.6977	31.7744
31.7744	31.7646	31.7701	31.7408	31.7909
31.7909	31.7744	31.7646	31.7701	31.7756
31.7756	31.7909	31.7744	31.7646	31.7756
31.7756	31.7756	31.7909	31.7744	31.7756
31.7756	31.7756	31.7756	31.7909	31.8226
31.8226	31.7756	31.7756	31.7756	31.8157
31.8157	31.8226	31.7756	31.7756	31.8203
31.8203	31.8157	31.8226	31.7756	31.8225
31.8225	31.8203	31.8157	31.8226	31.8231
31.8231	31.8225	31.8203	31.8157	31.8224
31.8224	31.8231	31.8225	31.8203	31.823
31.823	31.8224	31.8231	31.8225	31.8248
31.8248	31.823	31.8224	31.8231	31.8222
31.8222	31.8248	31.823	31.8224	31.8416
31.8416	31.8222	31.8248	31.823	31.8382
31.8382	31.8416	31.8222	31.8248	31.84
31.84	31.8382	31.8416	31.8222	31.84
31.84	31.84	31.8382	31.8416	31.8424

В якості типу функції приналежності вихідної змінної задамо лінійну функцію (рис. 3.104).

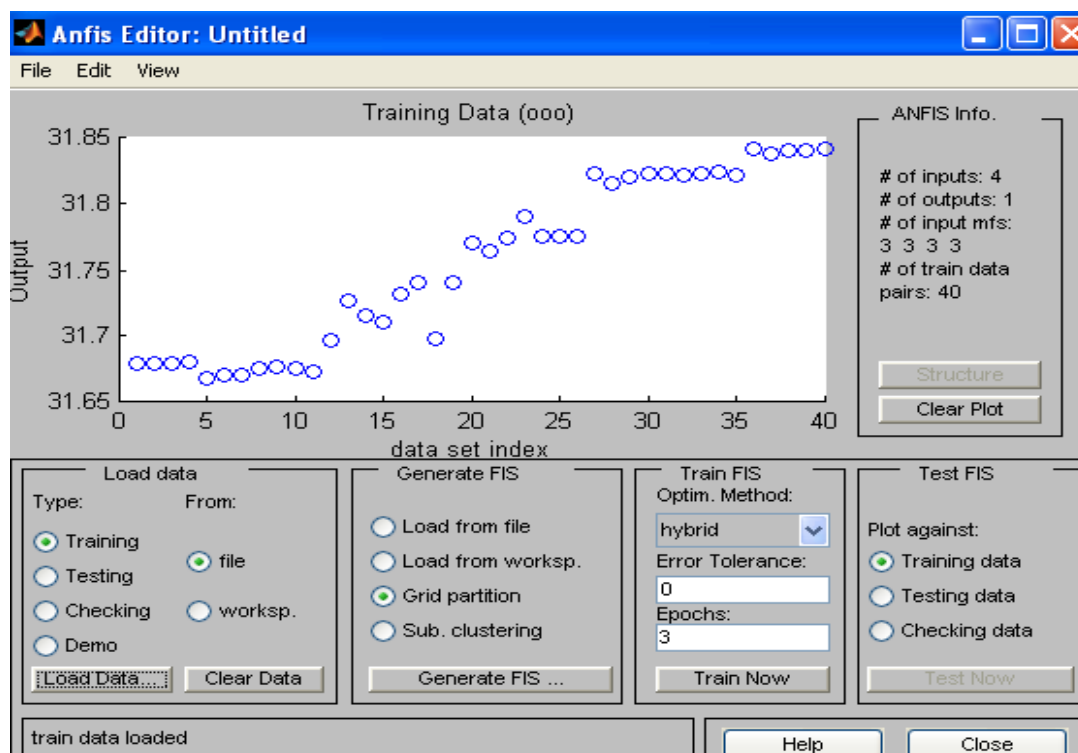


Рис. 3.103. Графічний інтерфейс редактора ANFIS після завантаження навчальних даних

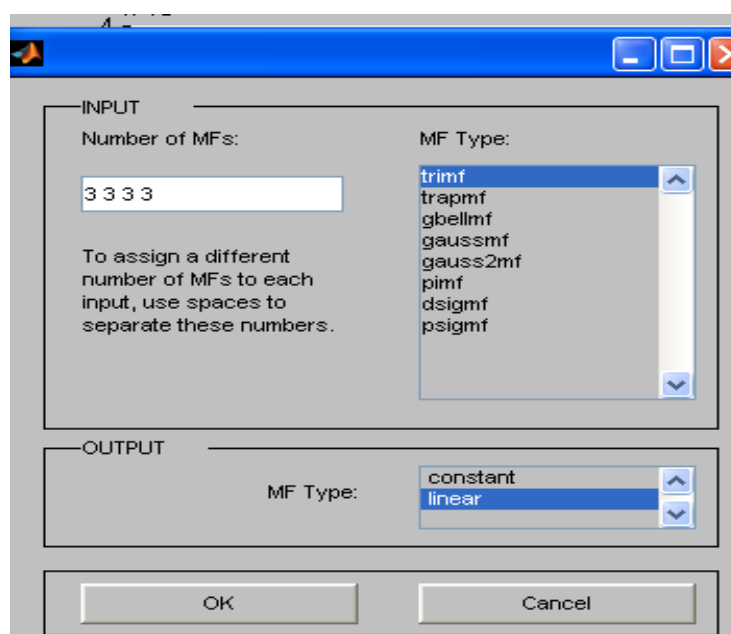


Рис. 3.104. Діалогове вікно для завдання кількості і типу функцій належності

Для навчання гібридної мережі скористаємося гібридним методом навчання з рівнем помилки Про, а кількість циклів навчання задамо рівним 10. Після закінчення навчання цієї гібридної мережі може бути виконаний аналіз графіка помилки навчання (рис.3.105), який показує, що навчання практично закінчилося після 3-го циклу.

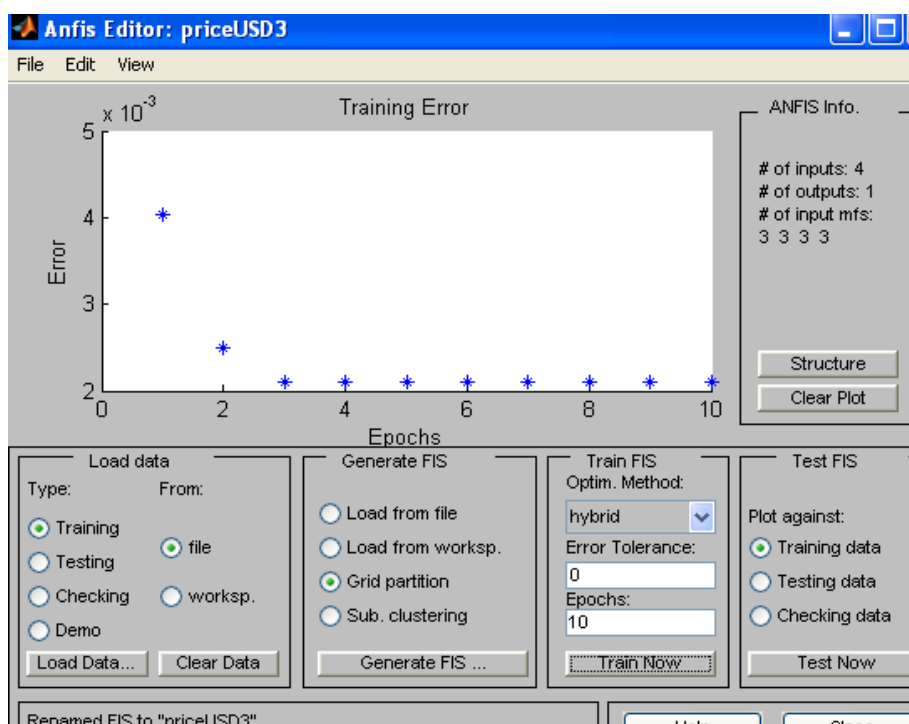


Рис.3.105. Графік залежності помилки навчання від кількості циклів навчання

Після навчання гібридної мережі можна візуально оцінити структуру побудованої нечіткої моделі (рис.3.106). Очевидно, графічна наочність даної моделі залишає бажати кращого, оскільки загальна кількість правил у розробленій адаптивній системі нейронечіткого виводу дорівнює 81, (що ускладнює їх візуальний контроль та оцінку).

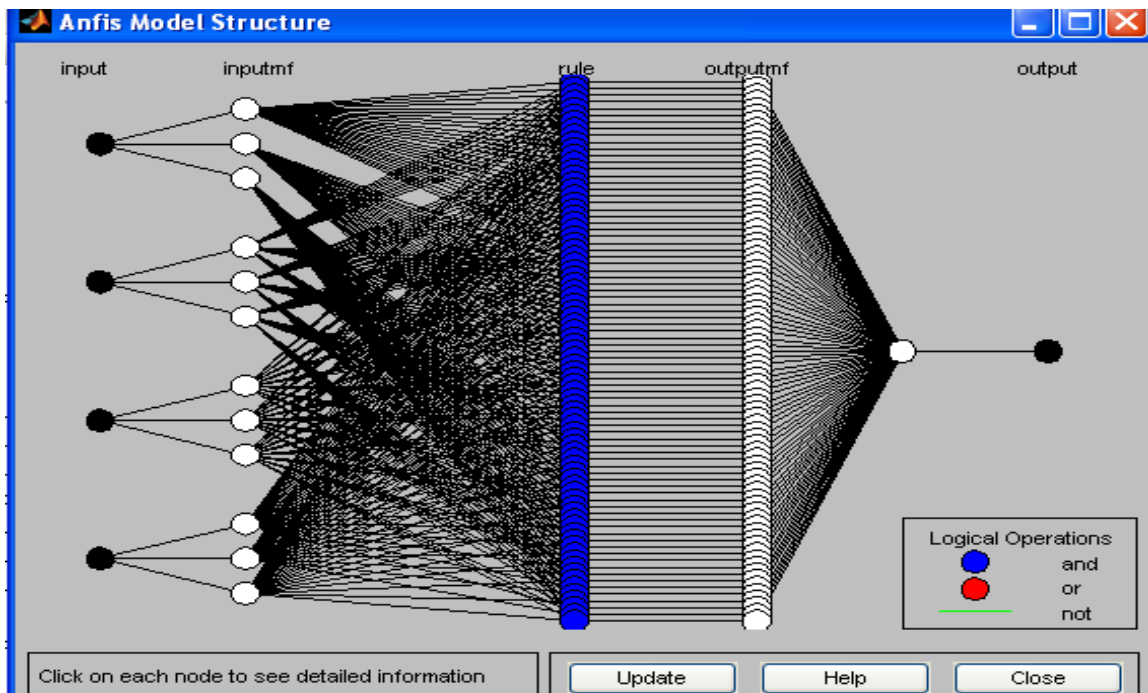


Рис.3.106. Структура згенерованої системи нечіткого висновку

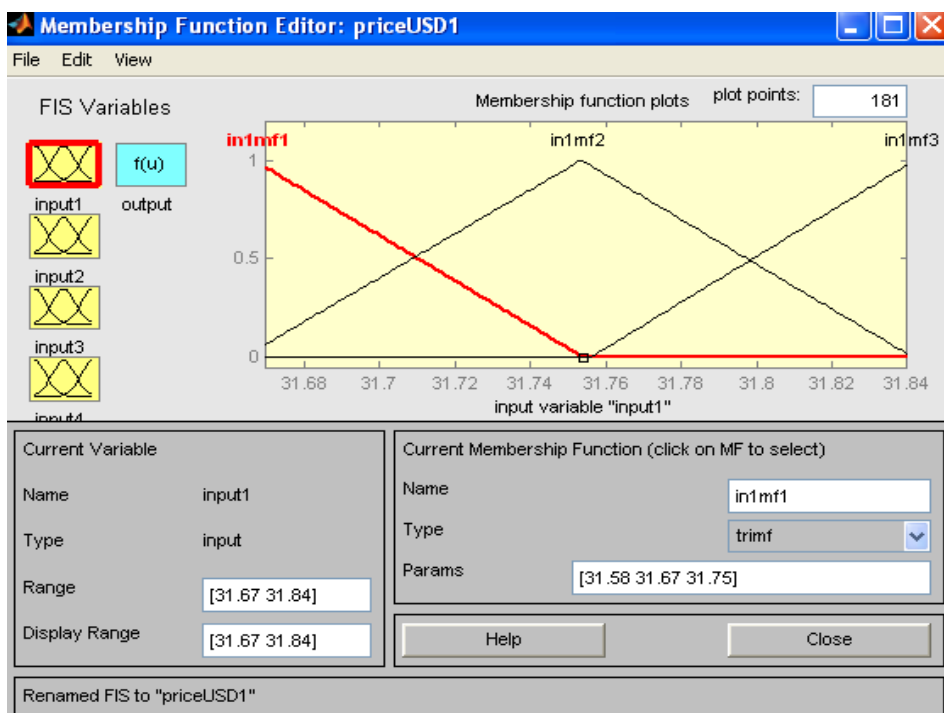


Рис. 3.107. Графічний інтерфейс редактора функцій приналежності побудованої системи нечіткого висновку для перевірки першої вхідної змінної

За допомогою графічних засобів системи MATLAB можна виконати контроль та налаштування параметрів функцій належності вхідних змінних і правил нечітких продукцій. Для виконання відповідних операцій можна скористатися редактором функцій належності (рис.3.107). Однак до перевірки адекватності побудованої нечіткої моделі залишимо всі параметри функцій належності без змін.

Виконаємо перевірку адекватності побудованої нечіткої моделі гібридної мережі. Для цієї мети зробимо ретроспективний прогноз значення курсової вартості USD на банківський день, наприклад, на 3 грудня 2002 р., вважаючи для цього випадку поточним банківським днем 30 листопада 2002. Оскільки точність кількісних значень, забезпечувана графічними засобами пакету Fuzzy Logic Toolbox, є недостатньою для вирішення даної задачі, скористаємося функцією командного рядка `evalfis`. В якості аргументів цієї функції вкажемо вектор значень курсової вартості USD на поточний і 3 попередніх банківських дня. Повний формат виклику цієї функції буде таким:

```
out = evalfis([31.8424 31.84 31.84 31.8382], priceUSD1),
```

де `out` – умовне ім'я вихідної змінної; `evalfis` – функція, що виконує виведення в системі FIS; 31.8424 – значення курсової вартості USD на 30.11.02; 31.84 – значення курсової вартості USD на 29.11.02; 31.84 – значення курсової вартості USD на 28.11.02; 31.8382 – значення курсової вартості USD на 27.11.02; `priceUSD1` – ім'я структури FIS, попередньо завантаженої в робочу область системи MATLAB.

```
>>out=evalfis([31.8424 31.84 31.84 31.8382],priceUSD1)
out
31.8547
```

Після виконання цієї команди за допомогою розробленої нечіткої моделі буде отримано значення вихідної змінної для 3.12.02, рівне 31.8547. Порівнюючи отримане значення з

відповідним значенням з табл. 3.4, можна констатувати збіг цих значень.

Таким чином, перевірка побудованої нечіткої моделі гібридної мережі показує досить високу ступінь її адекватності реальним вихідними даними, що дозволяє зробити висновок про можливість її практичного використання для прогнозування курсової вартості USD на фінансовому ринку валют. У цьому випадку нечіткі моделі адаптивних систем нейронечіткого висновку можуть вважатися новим і конструктивним інструментом технічного аналізу фінансових ринків.

Розглянутий підхід є перспективним напрямком для побудови і використання відповідних нечітких моделей прогнозування цін інших фінансових інструментів, таких як курси інших валют, акцій компаній, ф'ючерсів та опціонів. Дійсно, загальним для всіх цих інструментів з позицій технічного аналізу є відсутність апріорних припущень про динаміку коливань відповідних курсів цін, що цілком узгоджується з вихідними передумовами побудови нечітких моделей адаптивних систем нейронечіткого висновку.

3.11. Нечіткі експертні системи

Теорія нечітких множин, що, як очікувалося, перетворилася на методологію, розроблену в результаті досліджень штучного інтелекту, виявляє насамперед тісний зв'язок з експертними системами. Метод опису алгоритмів у формі "якщо.. те" був запропонований Заде у статті "Нечіткі алгоритми", опублікованої в 1968 р. Традиційні експертні системи не мали справ з нечіткостями, і ідея Заде полягала в тому, щоб розкрити сутність експертних систем шляхом опису алгоритмів словами, тобто нечітко, що виявилось дуже плідним. Як з'ясувалося пізніше, ця ідея допомогла вирішити проблему ненадійності в експертних системах. Безумовно, в експертних системах використовувалося

представлення невірогідності фактів і правил у виді коефіцієнтів упевненості так, як у системі діагностики інфекційних захворювань MYCIN. Операції з невірогідністю в цій системі засновані на ідеях байесовської імовірності. Імовірнісні висновки досліджуються як один з методів, використовуваних в експертних системах, але сутність експерта не у імовірнісній природі, а в нечітких тезах. Однак, говорячи про невірогідність у традиційних експертних системах, ми розуміємо під цим усього лише зовнішню оцінку фактів і правил, тобто не більш ніж різновид значень істинності в логіці. Наприклад, достовірність, чи ступінь істинності, твердження "сьогодні ранком накрапав дрібний дощ" розглядалася тільки зовні. Тут значення нечіткості слів "ранок", "дрібний дощ", "накрапав" внутрішньо взагалі не ставиться під сумнів. Насправді сутність недостовірностей в експертних системах полягає в нечіткості понять і слів, з якими оперує експерт. Подібні недостовірності не тільки не дозволяють зовні оцінити тези, але їх не можна обробляти, поки вони не введені в самі тези. Необхідно відповісти на питання: що такий ранок? Це 5 годин, 8 годин, а може, 10 годин? Достоїнство нечіткої логіки скоріше в чіткості невизначеності нечіткості сутності мови, а не в його логічності. При побудові моделі мислення експерта не можна абстрагуватися від ролі і сутності мови його думок. Завдяки оперуванню з невизначеностями слів і понять нечітка логіка надає методи, що наближають нас до сутності експертних систем. Якщо в процесі мислення людини за допомогою нестрогих міркувань є щось, називане логікою, то це не що інше, як нечітка логіка. Завдяки тому що в ній сполучаються нечіткості і логіка, вона може претендувати на роль мови опису в моделі мислення людини. Безсумнівно, експертні системи, будучи спроектованими як нечіткі експертні системи на базі нечіткої логіки, стануть більш ефективними і корисними. Реальним підтвердженням цього служать численні вже працюючі системи в області

керування. Майже всі експертні системи керування забезпечують нечітке керування, тобто є нечіткими експертними системами. Серед них чимало прикладів успішної автоматизації технологічних процесів за допомогою нечіткого керування. Відомі приклади застосування нечітких експертних систем для діагностики несправностей локомотивів, вибору меню в ресторанах і т. ін. Так що з погляду практичного використання число нечітких експертних систем уже перевищило число традиційних експертних систем.

В області **експертних систем** виділяють **три проблеми**: 1) представлення знань; 2) використання знань; 3) набуття знань.

Нечітка логіка може внести істотний вклад у рішення кожної з цих проблем. Для проблеми 1 головним виявляється те, що найбільш високий рівень представлення гарантований за допомогою мови; знання типу фактів, правил, оцінок й інші знання людини можна представити тільки у формі, що містить нечіткості. Більш того, оскільки людина допускає застосування невизначеностей, можна підібрати знання, що простіше використовувати. У представленні знань використовуються поняття нечітких множин, але для проблеми 2 ефективніше нечітка логіка. Основним методом є метод нечітких висновків. Особливість таких висновків полягає в одержанні висновків з нечіткої інформації за допомогою знань про нечіткі правила, але це ж можна сказати і про застосування знань експертами. Застосування знань не обмежується тільки висновками. Крім висновків серйозною проблемою є оцінки і судження. Як моделі оцінок і суджень експертів на основі емпіричних знань ефективні ідеї нечітких змін і нечіткого інтегрування.

При рішенні проблеми 3 у першу чергу треба навчитися витягати емпіричні знання експертів. При цьому частину таких знань можна передати тільки за допомогою нечітких слів, тому необхідна нечітка інтерпретація знань. Завдяки введенню нечіткої класифікації знань типу правил за умовами їхнього

застосування стає можливим гнучке набуття знань експерта. Другою за значенням задачею є пошук методів виймання знань, причому не тільки безпосередньо зі слів експерта, але і побічно, шляхом спостереження за діями експерта, чи навіть без експерта через експерименти і навчання. В останньому випадку для збору даних і інформації так, як це робить експерт, можна використовувати нечітку логіку. Іншими словами, виконують збір якісної інформації типу "якщо x мале, то y велике". Завдяки використанню нечіткого представлення можна визначити суть роботи експерта.

Крім керування існує безліч інших областей, у яких ефективно працюють нечіткі експертні системи. Серед них: пошук несправностей, медична діагностика, планування, проектування систем, економічні прогнози, інвестування цінних паперів, прогнозування збуту, прийняття економічних рішень, підготовка бібліографічних анотацій, консультації по законодавству тощо. Очікується поява нечітких експертних систем, що будуть використовуватися в "дорожніх" комп'ютерах для рішення задач малих підприємств та інших організацій.

Відзначимо, що експертні системи орієнтовані не на проблеми, формалізація яких проста і рішення яких можна знайти за допомогою математичних моделей і алгоритмів, тобто не на проблеми, з якими мають справу сучасні програми, а скоріше на складні проблеми, які можна вирішити тільки з використанням знань, отриманих експертами за довгі роки. Звичайно, у подібних випадках важко одержати знання, необхідні і достатні для рішення обраних проблем під час проектування системи. Такі знання підбирають методом проб і помилок, додаючи, чи змінюючи видаляючи знання. У цьому сенсі використання експертних систем у медицині має надзвичайно важливе значення.

Реляційні нечіткі експертні системи

У діагностичних нечітких експертних системах реалізуються спадні виводи, засновані на відшуканні прообразу нечіткої множини при відомому нечіткому відображенні (чи відношенні). У цьому випадку повний простір передумов X складається з m факторів (причин, відмовлень, дефектів і т. ін.), а повний простір висновків Y складається з n зовнішніх проявів (симптомів, ознак і т.і.):

$$X = \{x_1, x_2, \dots, x_m\},$$
$$Y = \{y_1, y_2, \dots, y_n\}.$$

Наприклад, у спрощеній моделі діагностики автомобіля:

x_1 – несправність акумулятора;

x_2 – відпрацьовування машинної олії;

y_1 – утруднення при запуску;

y_2 – погіршення вихлопу;

y_3 – недостатньо потужності.

При цьому між кожною парою елементів x_i і y_i існує причинний зв'язок, називаний нечітким відношенням (відповідністю, відображенням) R , що для кінцевих множин X і Y можна представити у вигляді матриці (r_{ij}) , елементи якої представляють ступінь причинних відносин у вигляді дійсного числа з інтервалу $[0,1]$.

Універсальна множину передумов X можна розглядати як вхід, а універсальну множину Y – як вихід нечіткої системи, модельованої відношенням R (рис.3.108).

Конкретні входи і виходи розглядаються як нечіткі множини A і B , побудовані відповідно на універсальних множинах X і Y . Знання експерта (у нашому випадку автомеханіка) формалізуються у вигляді матриці нечіткого відношення R , а зміст задачі складається у визначенні нечіткої множини причин

Аза відомої нечіткою множиною симптомів B , причому $B = A \circ R$.

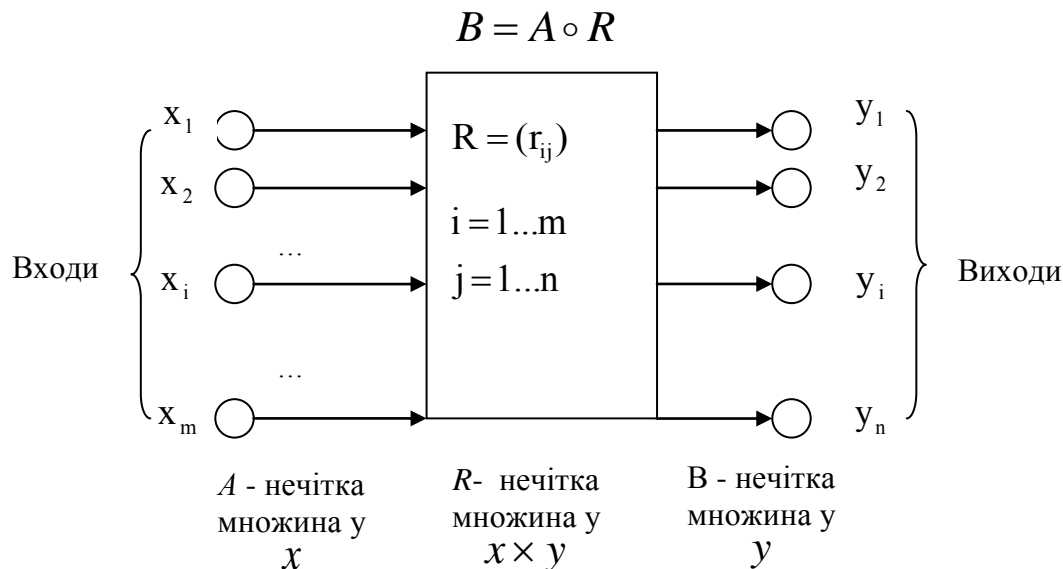


Рис. 3.108. Матриця нечіткого відношення

Повернемося до розглянутого прикладу. Нехай знання експерта мають вигляд:

$$R = \begin{matrix} & y_1 & y_2 & y_3 \\ \begin{matrix} x_1 \\ x_2 \end{matrix} & \begin{pmatrix} 0,9 & 0,1 & 0,2 \\ 0,6 & 0,5 & 0,5 \end{pmatrix} \end{matrix}$$

Якщо при огляді автомобіля виявилися труднощі при запуску, а потужність і вихлоп у нормі, то стан автомобіля можна оцінити, наприклад, як:

$$B = y_1 | 0,9 + y_2 | 0,1 + y_3 | 0,2.$$

Необхідно визначити причину такого стану.

$$A = x_1 | a_1 + x_2 | a_2$$

Входи і виходи можна представити у вигляді векторів-рядків:

$$B = (0,9; 0,1; 0,2);$$

$$A = (a_1; a_2).$$

Тоді, за умовою композиції відносин (A_iB розглядаються як унарні відносини)

$$(0,9 \quad 0,1 \quad 0,2) = (a_1 \cdot a_2) \circ \begin{pmatrix} 0,9 & 0,1 & 0,2 \\ 0,6 & 0,5 & 0,5 \end{pmatrix}$$

Використовуючи правило максимінного добутку матриць, маємо:

$$\begin{cases} 0,9 = (0,9 \wedge a_1) \vee (0,6 \wedge a_2), \\ 0,1 = (0,1 \wedge a_1) \vee (0,5 \wedge a_2), \\ 0,2 = (0,2 \wedge a_1) \vee (0,5 \wedge a_2). \end{cases}$$

Розв'язок цієї системи приводить до сукупності нерівностей. З першого рівняння маємо $a_1 \geq 0,9$. З другого $a_2 \leq 0,1$. Обидві ці умови задовольняють третьому рівнянню.

У підсумку одержуємо розв'язок:

$$\begin{cases} 0,9 \leq a_1 \leq 1, \\ 0 \leq a_2 \leq 0,1. \end{cases}$$

Таким чином, краще замінити акумулятор, чим змінювати олію. Варто звернути увагу на те, що можна запропонувати кілька рішень поставленої задачі, серед них максимальне $A_{\max} = (1;0,1)$ і мінімальне $A_{\min} = (0,9;0)$. У деяких випадках рішення відсутнє.

Використання нечітких множин в оцінці якості монотонних систем

Монотонними називаються системи, у яких погіршення якості роботи елементів приводить до погіршення якості роботи системи. Наприклад, у міру росту числа відмов елементів якість роботи такої системи може тільки погіршуватися. Хоча таку властивість мають не всі системи. Так, в обчислювальних системах і системах керування іноді спостерігаються ситуації, коли поява чергового відмовлення чи збою може компенсувати негативні наслідки попередніх відмовлень чи збоїв.

Структуру монотонної системи можна представити у вигляді деревоподібного графа, називаного деревом відмов чи деревом подій. Побудова такої мережі чи дерева являє собою окрему задачу, пов'язану з аналізом процесів чи подій, що протікають у розглянутій системі. Логіку подій (процесів) описують булевими функціями, що містять логічні зв'язки И, АБО, НЕ. Поєднання елементів у мережі в загальному випадку не зводиться до послідовно-рівнобіжного. Найпростішим прикладом такої мережі є місткова структура (рис. 3.109).

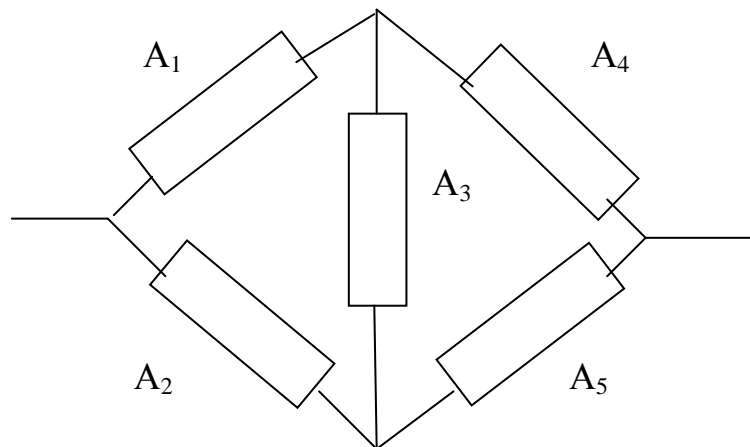


Рис.3.109. Місткова структура

Відзначимо, що електротехнічна аналогія тут є чисто умовною, що дозволяє зрозуміти структуру системи. Самі ж монотонні системи широко поширені не тільки в техніці, але й в управлінських науках (менеджементі), економіці, логістиці і т. ін. Аналізуючи логіку процесів, що протікають у мережі (наприклад, її провідність чи у загальному випадку "прохідність"), та її структуру, можна одержати множину мінімальних шляхів - набір мінімальних поєднань працездатних елементів, що забезпечують провідність мережі. Для нашого прикладу набір "прохідних" комбінацій представлений на рис.3.110.

Це означає, що мережа буде працездатною (проведеною), якщо працездатними (проведеними) є елементи: $(A_1 \text{ і } A_4)$ або $(A_2 \text{ і } A_5)$ або $(A_1 \text{ і } A_3 \text{ і } A_5)$ або $(A_2 \text{ і } A_3 \text{ і } A_4)$. Зворотня увага на повторюваність елементів у таких сполученнях, що визначає відмінність такої структури від звичайного послідовно-рівнобіжного поєднання. Перехід від рис.3.109 до рис.3.110 здійснюється у загальному випадку за допомогою перетворення структурних булевих функцій системи.

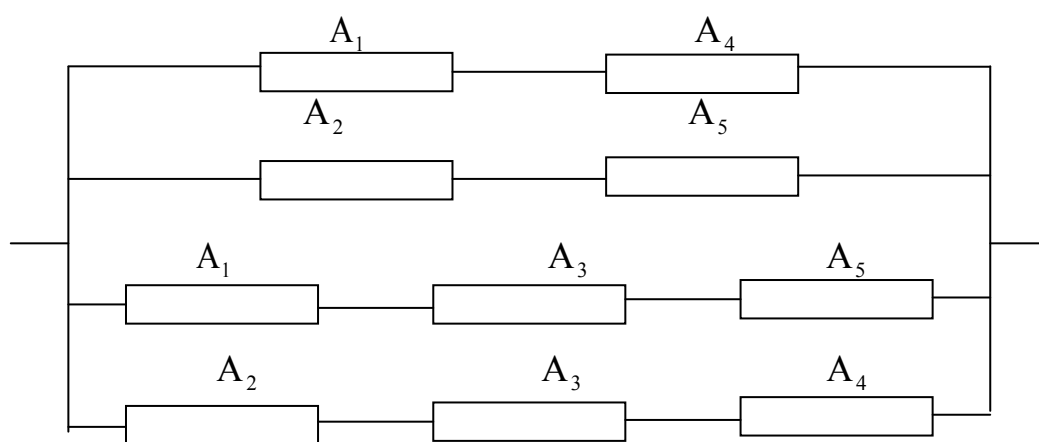


Рис. 3.110. Набір "прохідних" комбінацій

У розглянутому прикладі провідності мережі з роботою кожного елемента A_i може бути пов'язана бінарна змінна z_i , приймаюча лише два значення, наприклад, $z_i = 0$, якщо елемент не працює, і $z_i = 1$, якщо елемент працює.

У багатьох системах техніки, економіки, керування в процесі роботи систем відбуваються не відмови елементів (повна втрата працездатності), а часткове погіршення якості їхнього функціонування (т.зв. функціональні відмови). Наприклад, у великих енергосистемах спадання напруги на окремих ділянках мережі звичайно приводить не до повної відмови всієї системи, а тільки лише до погіршення якості її роботи. Неважко

представити таку ситуацію в логістиці, скажемо, у роботі системи керування запасами і т. ін.

Нечіткий підхід до оцінки якості монотонних систем

У більш широкій постановці задачі аналізу якості роботи монотонної системи з роботою кожного елемента мережі можна зв'язати нечітку величину (нечітка підмножина), що характеризує якість роботи елемента. У найпростішому випадку ці нечіткі величини можна вважати невзаємодіючими, однак, у принципі, можливе й урахування взаємодії. Задачею аналізу є визначення якості функціонування системи залежно від нечітко описаної якості функціонування елементів.

Нехай, наприклад, універсальну множину оцінок X визначено в такий спосіб: $X = \{x_1, x_2, x_3, x_4, x_5\}$, де

$$x_1 = \{\text{працює відмінно}\},$$

$$x_2 = \{\text{працює дуже добре}\},$$

$$x_3 = \{\text{працює добре}\},$$

$$x_4 = \{\text{працює погано}\},$$

$$x_5 = \{\text{не працює}\},$$

На цій універсальній множини X експертним методом побудуємо нечітку множину A_i , що характеризує якість роботи елемента A_i :

$$A_i = \{(x_1 | p_{i1}), \dots, (x_5 | p_{i5})\}.$$

Структура, утворена рівнобіжним поєднанням елементів (рис.3.111), працює з показником якості x_i в тому випадку, якщо з цим показником якості працює хоча б один з елементів A чи B , тобто якість роботи такої структури описується нечіткою множиною, що є об'єднанням нечітких множин A і B , пов'язаних з елементами A і B .

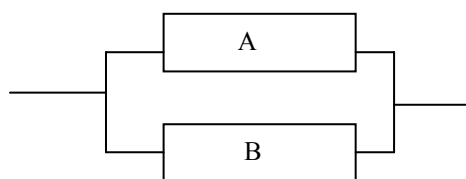


Рис. 3.111. Паралельне з'єднання елементів

Структура, утворена послідовним з'єднанням елементів (рис.3.112), працює з показником якості x_i у тому випадку, якщо з цим показником якості працюють обидва елементи A і B , тобто якість роботи такої структури описується нечіткою множиною, що є перетинанням нечітких множин A і B , зв'язаних з елементами A і B .

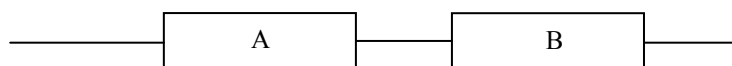


Рис.3.112. Послідовне з'єднання елементів

Нехай нечіткі множини A_1, \dots, A_5 мають вигляд, наведений у таблиці 3.8, і відповідають елементам за рис.3.109 і 3.110.

Таблиця 3.8. Нечіткі множини A_1, \dots, A_5 , відповідні елементам A_1, \dots, A_5 за рис. 3.109 і 3.110

X	$\mu_{A1}(x)$	$\mu_{A2}(x)$	$\mu_{A3}(x)$	$\mu_{A4}(x)$	$\mu_{A5}(x)$
x_1	0,1	0,2	0,3	0,6	0,3
x_2	0,3	0,7	0,2	0,7	0,2
x_3	0,4	0,6	0,1	0,8	0,1
x_4	0,5	0,5	0,6	0,9	0,3
x_5	0,9	0,1	0,7	0,7	0,6

Результати операцій над множинами A_1, \dots, A_5 представлені в табл. 3.9.

Таблиця 3.9. Результати операцій над множинами A_1, \dots, A_5

X	$\mu_{B_1}(x)$ ($B_1 = A_1 \cap A_4$)	$\mu_{B_2}(x)$ ($B_2 = A_2 \cap A_5$)	$\mu_{B_3}(x)$ ($B_3 = A_1 \cap A_3 \cap A_5$)	$\mu_{B_4}(x)$ ($B_4 = A_2 \cap A_3 \cap A_4$)	$\mu_K(x)$ ($K = B_1 \cup B_3 \cup B_4$)
x_1	0,1	0,2	0,1	0,2	0,2
x_2	0,3	0,2	0,2	0,2	0,3
x_3	0,4	0,1	0,1	0,1	0,4
x_4	0,5	0,3	0,3	0,3	0,5
x_5	0,7	0,1	0,6	0,1	0,7

Функція приналежності $\mu_K(x)$ множини K

$$K = \{x_1 | 0,2; x_2 | 0,3; x_3 | 0,4; x_4 | 0,5; x_5 | 0,7\}$$

нечітко описує якість функціонування системи зі структурою за рис. 3.109 при заданій якості функціонування елементів (табл.3.8), визначеній на універсальній множині оцінок X . Для цілей керування множину можна дефазифікувати.

3.12. Програмні пакети в області нечіткої логіки

Для вирішення завдань із застосуванням систем нечіткої логіки існує достатня кількість програмних продуктів, що розрізняються ступенем складності і наданими можливостями. Зазначимо деякі з них.

Пакет CubiCalc розроблений американською фірмою Нурел-Logis на початку 90-х рр. ХХ ст. Слід зазначити, що до 1995 р. експорт цього програмного продукту в Росію був заборонений американським Комітетом з контролю за експортом (СОСОМ), так як системи з НЛ використовувалися в американській програмі «Стратегічної оборонної ініціативи». Включення зараз до пакету в якості ілюстративного прикладу завдання про собаку, який наздоганяє kota, є не що інше, як завдання про ракету і протиракету. Використовуючи обмежений набір вхідних змінних (координати собаки і kota, кут між напрямками їх рухів) і єдину

вихідну змінну (величину кута переміщення собаки), базу знань, що складається всього з п'яти правил, програма успішно вирішує поставлене завдання. Після того як «собака», яка швидко біжить, стала наздоганяти міжконтинентального «кота», скептицизм, який виявляють противники НЛ, різко зменшився, а системи з НЛ були включені в оборонні програми США. Сьогодні CubiCalc застосовується при вирішенні різних завдань: від адаптивного управління оптовими складами до моделювання ринку ф'ючерсних контрактів. Цей пакет містить засоби для введення і представлення даних, формування правил виведення, опису нечітких множин. До деяких версій пакету включається модуль RuleMaker, що дозволяє вирішувати одну з основних проблем у системах з НЛ: автоматична побудова нечітких правил. Пакет дозволяє контролювати кожен крок обчислень, генерувати на мові Сі тексти, що містять алгоритми роботи нечіткої системи, які потім можуть бути вбудовані в додатки користувача.

Процес виконання будь-якого проекту в пакеті CubiCalc складається з декількох кроків: – ініціалізація; ввід даних; попередня обробка; виконання правил; постобробка; виведення даних; моделювання процесу.

Модуль Fuzzy Logic Toolbox з системи Matlab. Цей модуль являє собою пакет розширення системи Matlab, яка була розроблена американською фірмою MathWorks для технічних додатків. Одним з достоїнств системи Matlab є можливість її розширення з метою вирішення нових, виникаючих завдань. Описуваний тут модуль Fuzzy Logic Toolbox використовується для вирішення завдань в області НЛ. Основні можливості цього пакету: побудова систем нечіткого виводу (експертних систем, апроксиматорів залежностей, різних регуляторів); формування адаптивних нечітких систем (гібридних нейронних мереж); інтерактивне динамічне моделювання.

Як описувалося вище, рішення задач на основі НЛ зводиться до виконання ряду послідовних дій, зокрема: вибір вхідних та

вихідних змінних, формування бази правил для оцінюваної системи, побудова вихідної нечіткої множини і її дефазифікації з метою отримання чіткого виходу системи. Пакет Fuzzy Logic Toolbox дозволяє виконати наведені операції, графічно відображаючи проміжні і підсумкові результати.

Пакет WINROSA. Розроблено в Німеччині на початку 90-х рр. Робота з цим пакетом являє собою таку послідовність кроків: визначення проекту; формування правил; зменшення кількості правил; аналіз результатів; експорт результатів.

Розглянемо ці кроки більш докладно.

Для роботи з цим пакетом спочатку необхідно визначити вхідні і вихідні змінні. Вхідні змінні представляють собою передумови правил «якщо ..., то», а вихідні – укладення цих правил. Необхідно мати результати спостережень для кожної змінної. Лінгвістичні змінні і ФП повинні бути визначені перед формуванням правил. На другому кроці користувач повинен виділити максимальне число значущих лінгвістичних співвідношень, що дозволяє оцінити величину простору пошуку. При цьому необов'язково узагальнювати в одному правилі все вхідні змінні. Для малих просторів пошуку можливий метод повного перебору, для великих – необхідно застосовувати еволюційні процедури пошуку. На цьому етапі позбуваються правил, які повністю перекриваються кращими правилами. Крім того, кожному правилу надають вагу (між нулем та одиницею), яка показує, як добре дане правило підтверджується наявними даними. Тут формуються не тільки позитивні правила, які виражають рекомендації, але і негативні, що визначають заборони і застереження.

Результатом другого етапу є множина прийнятних правил. Так як вона складається з узагальнюючих правил, можуть мати місце різні перекриття правил і навіть конфлікти між останніми. Хоча подібні явища вирішуються при дефазифікації, рекомендується на третьому кроці зменшити, наскільки це

можливо, кількість використовуваних правил. Це призводить до більшої прозорості вирішення і скорочення часу обчислень.

На четвертому кроці при аналізі результатів правила повинні бути перевірені на правдоподібність. При цьому важливу роль відіграє вага правила. Високі значення цієї величини свідчать про явну залежність між змінними, у той час як малі – про суперечливість змінних у даному правилі. Для отримання швидкого огляду бази правил корисно застосувати графічне представлення. При розгляді, наприклад, всіх правил, які містять певну вхідну змінну, важливість цієї величини може бути знайдена з графічного відображення. У багатьох додатках необхідно, щоб, принаймні, було одне правило для кожної вхідної ситуації.

При експорті результатів необхідно враховувати, що файл WINROSA зберігається як ASCII-файл з розширенням «* .ini». Для прямого використання результатів в імітаційних завданнях експортовані файли записані для DORA for Windows 6.0; Matlab і fuzzyTECH.

Пакет *FIDE* (Fuzzy Inference Development Environment) розробки американської фірми Aptronix є засіб для створення і використання нечітких систем виведення. Ядром цієї системи є компілятор, який застосовує англійський синтаксис і позначення для змінних, ФП і правил. В якості входу система використовує текстовий формат, що становить значну гнучкість для користувача. Крім того, розробник може скористатися будь-яким текстовим редактором, включаючи і власний редактор системи *FIDE* для створення і зміни правил. До складу пакету входить матричний редактор правил з метою більш ефективного розробки правил і графічний редактор відображення ФП. Система *FIDE* також забезпечує повним набором засобів візуалізації і налагодження. Користувач має можливість переглянути розроблювану систему нечіткого висновку у тривимірному просторі або в будь-якому перетині цього простору. Після того як

розроблена окрема комірка нечіткого виводу, що складається з вхідних, вихідних змінних і набору правил, користувач може з'єднати їх з елементами, виконаними поза системою FIDE, для формування інтегрованої структури. FIDE має у своєму складі спеціальний засіб, званий «Composer» (упорядник), який символічно відображає окремі осередки нечіткого виведення і дозволяє розробнику графічно з'єднувати входи і виходи кожного осередку. Такий підхід дає можливість здійснювати моделювання системи і графічно розглядати її поведінку.

Демонстраційна версія системи FIDE знаходиться за адресою: <http://www.aptronix.com>.

З досвіду роботи з пакетами CubiCalc і Fuzzy Logic системи Matlab – краще остання, унаслідок великих можливостей у частині вибору виду ФП, композиції нечітких висновків і графічного відображення результатів.

Розділ 4

ГЕНЕТИЧНІ АЛГОРИТМИ

Генетичні алгоритми застосовуються для рішення оптимізаційних задач за допомогою методу еволюції, тобто шляхом добору з безлічі рішень найбільш підходящого. У типовій задачі оптимізації існує набір змінних, що впливають на процес, і формула або алгоритм, які використовують ці змінні для побудови моделі цього процесу. При цьому завдання полягає в тому, щоб знайти такі значення змінних, які певним чином оптимізують модель. У разі, якщо моделлю є формула, то зазвичай відшуковують максимум або мінімум функції, яку дана формула представляє. Існує багато математичних методів, які вирішують завдання оптимізації в тому випадку, якщо це завдання з «гарною поведінкою». Однак традиційні методи терплять крах, якщо завдання не належить до цього класу. Є великий клас задач, для яких обчислювальні труднощі ростуть експоненціально з розмірністю задачі. Прикладами завдань з такою «поганою поведінкою» можуть служити комбінаторні задачі, а також задачі, математичний опис яких не є гладкою безперервною функцією. Тут мінімізується цільова функція (цільова функція, функція вигоди або функція вартості), яка залежить від порядку кінцевого числа об'єктів. Кількість варіантів розміщення N об'єктів, і, отже, зусиль для знаходження мінімуму цільової функції збільшуються в геометричній прогресії із зростанням N .

4.1. Сутність еволюційних обчислень

В останні 30 років з'явився інтерес до завдань, вирішення яких ґрунтується на принципах еволюції та успадкування ознак. Системи подібного роду містять популяцію потенційних рішень,

мають певний процес відбору, який використовує критерії придатності індивідуумів (окремих рішень), застосовують деякі оператори рекомбінації. До таких систем відноситься клас еволюційних обчислень (ЕО). Під останніми розуміється термін, використовуваний для опису алгоритмів пошуку, оптимізації або навчання, заснованих на деяких формальних ознаках природного еволюційного відбору [25]. Методи ЕО часто застосовуються для опису процесів еволюції програм або функцій (генетичне програмування), кінцевих автоматів (еволюційне програмування) і систем, заснованих на продукційних правилах (класифікаційні системи). Іноді ЕО разом з НЛ (нечітка логіка) використовуються для навчання нейронних мереж, що призвело до нового терміну "м'які обчислення", що об'єднали ГА, НМ і ШНМ.

Серед методів ЕО можна виділити такі:

еволюційне програмування – представляє рішення задачі у вигляді універсальних кінцевих автоматів, які реагують на подразнення з зовнішнього середовища;

еволюційні стратегії – кожне рішення знаходиться у вигляді масиву числових параметрів, що визначають аргумент цільової функції;

генетичні алгоритми – кожне рішення є бітовим рядком (хромосомаю) певної довжини в популяції фіксованого розміру;

генетичне програмування – тут застосовуються ідеї ГА для еволюції комп'ютерних програм.

В самому загальному вигляді метод ЕО можна таким чином. Метод ЕО являє собою імовірнісний алгоритм, який містить популяцію індивідуумів $P(t) = \{x_1^t, x_2^t, \dots, x_n^t\}$ на ітерації t . Кожен індивідуум є потенційним рішенням розглянутої проблеми, і в будь-якому з методів ЕО реалізується як деяка (можливо, складна) структура даних S . Кожне рішення x_i^t оцінюється для отримання величини придатності (в англійській мові часто використовується термін "fitness"). Потім створюється нова популяція

(ітерація $t+1$) шляхом відбору найбільш придатних індивідуумів (етап селекції).

Деякі члени нової популяції зазнають перетворень (етап рекомбінації) за допомогою генетичних операторів для формування нових рішень. Серед операторів можна виділити оператор мутації, який створює новий індивідуум шляхом малих змін вихідного, і оператор схрещування, формує нові індивідууми допомогою комбінування частин кількох вихідних. Після ряду таких генерацій програма сходиться, тобто знаходиться кращий індивідуум, який представляє собою оптимальне рішення.

Очевидно, що багатоеволюційні програми (синонім еволюційних обчислень) можуть бути сформульовані для вирішення конкретної проблеми. Такі програми можуть мати різну структуру даних для побудови окремого індивідуума, генетичні оператори для трансформації індивідуумів, методи для створення початкової популяції, параметри обчислень (розмір популяції, ймовірності застосування різних операторів і т.ін.).

Слід зазначити, що методи ЕО не гарантують виявлення глобального оптимуму за прийнятний час. Практичний інтерес до них пояснюється тим, що ці методи дозволяють знайти «хороші» рішення дуже складних завдань за менший час, ніж за допомогою інших методів.

Необхідно усвідомити різницю між еволюційним програмуванням і ГА: при першому підході використовується будь-яка структура даних (хромосомне подання), підходить для розв'язання задачі, і будь-яка множина генетичних операторів; у другій ситуації застосовуються двійкові рядки фіксованої довжини (хромосоми) і два оператори (бінарні мутація і схрещування).

В еволюційних програмах проблема обмежень має інший характер. Тут мова йде, скоріше, про вибір «кращого» хромосомного представлення шуканого рішення разом зі значущими генетичними операторами для задоволення всіх обмежень, що накладаються цією проблемою. Будь-який генетичний оператор пови-

нен проходити через деяку характеристичну структуру від «батьків» до «нащадків», тому структура подання відіграє важливу роль у визначенні генетичних операторів. Крім того, різні структури уявлень мають різні характеристики придатності для обмежень, що ще більше ускладнює завдання. Ці два компоненти – представлення та оператори – впливають один на одного.

4.2. Основні поняття генетичних алгоритмів

Ідея використання ГА полягає в тому, щоб виконувати те, що робить природа. Розглянемо один приклад з популяцією кроликів, наявною у деякий момент часу t . Частина з них більш швидка і моторна, ніж інша. Такі швидкі кролики мають менше шансів бути з'їденими лисицями, і тому більшість з них виживає, щоб робити те, що кролики добре роблять: створювати ще більше кроликів. Вижила популяція кроликів дає нове потомство, яке має гарну суміш генетичного матеріалу: деякі "повільні" кролики схрещуються з "швидкими", швидкі – з швидкими, моторні – з незграбними і т. ін. Результуюче потомство буде у середньому більш швидким і спритним, ніж у вихідній популяції, тому що тільки такі особини мають шанс уникнути загибелі.

Генетичні алгоритми виконують крок за кроком процедуру, яка пов'язана з прикладом з життя кроликів. Отже, одним з можливих визначень може служити таке: ГА – це алгоритми, методи пошуку, що моделюють приватні природні явища: генетичну спадковість і боротьбу за виживання.

Перш ніж перейти до структурі ГА, коротко висвітлимо історію генетики. Генетика – це вивчення спадковості. Генетики вивчають взаємодія генів та їх передачу від батьків до потомства. У живих організмах клітини містять важливі групи спеціальної матерії, що містить спадкову інформацію. Такі ниткоподібні структури називаються хромосомами. Гени – спеціальні елементи, які переносяться хромосомами. Основний процес кодування

інформації в межах генів і хромосом досить простий, але можливі результати майже безмежні за кількістю. Генетична структура кожного організму називається генотипом, який визначає і обмежує багато аспектів розвитку і відбору. Відгук організму на зовнішні умови принципово визначається його **генотипом**. Властивості організму, які виникають з цього зіткнення з навколишнім середовищем, визначають його **фенотип**.

Основний принцип природного відбору як головного принципу еволюції був сформульований Ч. Дарвіном задовго до відкриття генетичних механізмів. Ігноруючи основні спадкові принципи, Ч. Дарвін висунув гіпотезу злиття, вважаючи, що батьківські якості змішуються разом, подібно до рідин, на досліджуваному організмі. Його селекційна теорія викликала серйозну критику з боку інших науковців, проте в той час ніякої більш прийнятної генетичної теорії ще не було.

Значний крок уперед у період 1856–1863 рр. в розробці генетики зробив католицький священик з Чехії Р. Мендель (G. Mendel), який відкрив основні принципи передачі спадкових факторів від батьків до потомства. У результаті багаторічних досліджень він відкрив закони домінування ознак у першому поколінні незалежно від їх розподілу в наступних поколіннях і їх кількісного співвідношення.

Генетика була повністю розроблена американським вченим Т. Морганом (T. Morgan) у період 1909–1914-х рр. і його колегами, які експериментально довели, що хромосоми є головними носіями спадкової інформації, а гени, які представляють спадкові фактори, розташовані в лінію на хромосомах. Т. Моргану вдалося спостерігати в мікроскоп процес обміну генетичного матеріалу між різними хромосомами: дві хромосоми зближувалися і схрещувалися, обмінюючись фрагментами. Він представляв гени впорядкованими по довжині хромосом, як намистинки намисто. Експериментальні дані привели його до чудової ідеї про створення генетичних карт. Очевидно, що чим далі знаходяться два гена

один від одного, тим більше ймовірність обриву нитки, яка їх зв'язує, й отримання нових поєднань генів.

На початку 1960-х рр. деякі біологи почали експериментувати з комп'ютерною імітацією генетичних систем. Зазначимо, що сучасна теорія ГА, застосовна для вирішення оптимізаційних завдань, пов'язана з ім'ям Д. Холланда.

Генетичні алгоритми використовують словник, запозичений з природної генетики. Тут необхідно відзначити все ще не установлену термінологію в цій області, і тому в різних виданнях можна зустріти різну інтерпретацію однакових понять. Нижче будемо дотримуватися наступної термінології:

бітовий рядок 0101...101 (хромосома, індивід, індивідуум) – визначає точку простору пошуку та представляє потенційне рішення задачі;

гени – елементи, з яких складається хромосома (синонім генів – ознаки, букви);

популяція – набір хромосом (рядків); у класі ГА розмір (величина) популяції приймається фіксованою величиною;

покоління (генерація) – нове покоління після кожного кроку роботи ГА;

батьки – хромосоми, з яких шляхом схрещування і відбору формуються хромосоми "нащадки";

якість хромосоми – функція, що оцінює придатність даного рядка порівняно з іншими (в англ. мовою – fitness). генетичні оператори (відбір, схрещування, мутація) – перетворення, яким піддаються хромосоми в процесі еволюції і боротьби за виживання.

Підкреслимо, що кожний рядок (хромосома) представляє потенційне рішення задачі. Еволюційний процес через популяцію хромосом відповідає пошуку по простору потенційних рішень.

Популяція здійснює імітаційну еволюцію: при кожній генерації відносно "хороші" рішення відтворюються, в той час як "погані" – помирають. Для відмінності між поганими і хорошими

рішеннями використовується оціночна функція, визначальна якість кожного рядка.

Для того щоб виявити відмінність ГА при пошуку оптимальних рішень від традиційних способів вирішення оптимізаційних задач, коротко вкажемо принцип дії двох загальноприйнятих методів: **підйому на пагорб** і **імітаційний відпал**.

Метод підйому на пагорб (hillclimbing method) використовує ітераційний поліпшуючий підхід. Метод застосуємо до єдиної (поточної) точки в просторі пошуків. Під час однієї ітерації наступна точка вибирається з умови сусідства з поточною точкою. У випадку, якщо нова точка забезпечує кращу величину цільової функції, то вона стає поточною. В іншому випадку вибирається інша сусідня точка, яка порівнюється з поточною. Метод пошуку оптимуму завершується, якщо подальшого поліпшення в поведінці цільової функції не настає. Цей метод забезпечує лише локальний оптимум, а тривалість пошуку залежить від вибору початкової точки. Крім того, тут відсутня інформація про відносну помилку (по відношенню до глобального оптимуму) знайденого рішення. Для збільшення шансів на успіх метод зазвичай виконується для значного числа початкових точок.

Імітаційний відпал (simulated annealing) заснований на ідеї, запозиченій зі статистичної фізики. Цей метод описує поведінку матеріального тіла при твердінні із застосуванням процедури відпалу (керованого охолодження) при температурі, послідовно зменшуваної до нуля. У реальних процесах кристалізації твердих тіл температура знижується ступінчастим чином. На кожному рівні вона якийсь час підтримується постійною, що необхідно для забезпечення термічної рівноваги. Протягом усього періоду часу, коли температура залишається вище абсолютного нуля, вона може як підвищуватися, так і знижуватися. За рахунок утримання температури процесу поблизу від значення, відповідного рівню термічної рівноваги, вдається обходити пастки локальних максимумів. Метод імітації відпалу являє собою алгоритмічний аналог

фізичного процесу керованого охолодження. У даний час даний метод вважається одним з небагатьох алгоритмів, що дозволяють практично знаходити глобальний мінімум функції декількох змінних.

В описі алгоритму в якості назви параметра, що впливає на ймовірність збільшення значення цільової функції, використовується термін "температура", хоча з формальної точки зору наведена модель оптимізації є тільки математичною аналогією процесу відпалу. **Алгоритм імітації відпалу** виглядає концептуально нескладним і логічно обґрунтованим. Насправді доводиться вирішувати багато фундаментальних проблем, які впливають на його практичну придатність. Основною слід назвати проблему тривалості імітації. Для підвищення ймовірності досягнення глобального мінімуму тривалість відпалу (подається кількістю циклів, повторюваних при одному і тому ж значенні температури) повинна бути досить великою, а коефіцієнт зменшення температури – низьким. Це збільшує тривалість процесу моделювання, що може дискредитувати його з позицій практичної доцільності.

Метод імітаційного відпалу виключає більшість недоліків методу підйому на пагорб: рішення не залежить від початкової точки і зазвичай є близькою до оптимальної точки. Це досягається введенням ймовірності p заміни поточної точки новою точкою $p = 1$, якщо нова точка забезпечує краще значення цільової функції; однак $p > 0$ у протилежній ситуації. В останньому випадку ймовірність p залежить від значень цільової функції для поточної і нової точок та керуючого параметра температури. Під час роботи методу температура знижується по кроках, і алгоритм завершується при деякому малому значенні температури.

Відмінності ГА від традиційних методів пошуку полягають у такому:

– для пошуку оптимуму використовують кілька точок одночасно, а не переходять від точки до точки, як це робиться у традиційних методах, що дозволяє уникнути небезпеки попадання в

локальний екстремум цільової функції, якщо вона не є унімодальною, тобто має кілька таких екстремумів. Використання декількох точок одночасно значно знижує таку можливість;

– в процесі роботи ГА не вимагають ніякої додаткової інформації, що збільшує швидкість роботи алгоритму (єдина інформація: область допустимих значень і функція придатності в певних точках);

– використовують і детерміновані правила для переходу від одних до інших точок, і ймовірнісні правила для породження нових точок аналізу;

– працюють з кодами, в яких представлений набір параметрів, що залежать від аргументів цільової функції. Інтерпретація цих кодів відбувається тільки перед початком роботи алгоритму і після завершення її роботи для отримання результату. У процесі роботи маніпуляції з кодами відбуваються абсолютно незалежно від їх інтерпретації, код розглядається просто як бітовий рядок.

Структура ГА така ж, як і в будь-якій еволюційній програмі. Під час ітерації t ГА містить популяцію потенційних рішень (хромосом, векторів) $P(t) = \{x_1^t, \dots, x_n^t\}$. Кожне рішення x_i^t оцінюється для того, щоб отримати деяку міру її придатності (fitness). Потім на ітерації $t + 1$ формується нова популяція шляхом відбору найбільш придатних індивідуумів. Деякі члени цієї нової популяції піддаються репродукції за допомогою генетичних операторів: схрещування та мутації для утворення нових рішень. Схрещування поєднує ознаки двох батьківських хромосом для утворення двох нащадків обміном відповідних сегментів батьків. Наприклад, якщо батьки відображаються п'ятивимірними векторами $(a_1, b_1, c_1, d_1, e_1)$ та $(a_2, b_2, c_2, d_2, e_2)$, то, вибравши точку схрещування після другого гена, отримуємо таких нащадків: $(a_1, b_1, c_2, d_2, e_2)$ та $(a_2, b_2, c_1, d_1, e_1)$.

Ясно, що оператор схрещування являє собою обмін інформацією між різними потенційними рішеннями. Мутація довільно змінює один або більше генів обраної хромосоми випадковою за-

міною значення, що дорівнює одиниці, на нуль або навпаки. Очевидно, що оператор мутації вводить деяку варіативність у популяції хромосом. ГА для вирішення будь-якої проблеми повинен містити, як правило, такі компоненти:

- генетичне подання потенційних рішень задачі;
- спосіб створення початкової популяції потенційних рішень;
- оціночну функцію, яка відіграє роль оточення і ранжує рішення щодо ступеня їх придатності;
- генетичні оператори, змінюють генетичний склад потомства;
- значення параметрів ГА (ймовірність схрещування і мутації, розмір популяції, кількість поколінь та ін.).

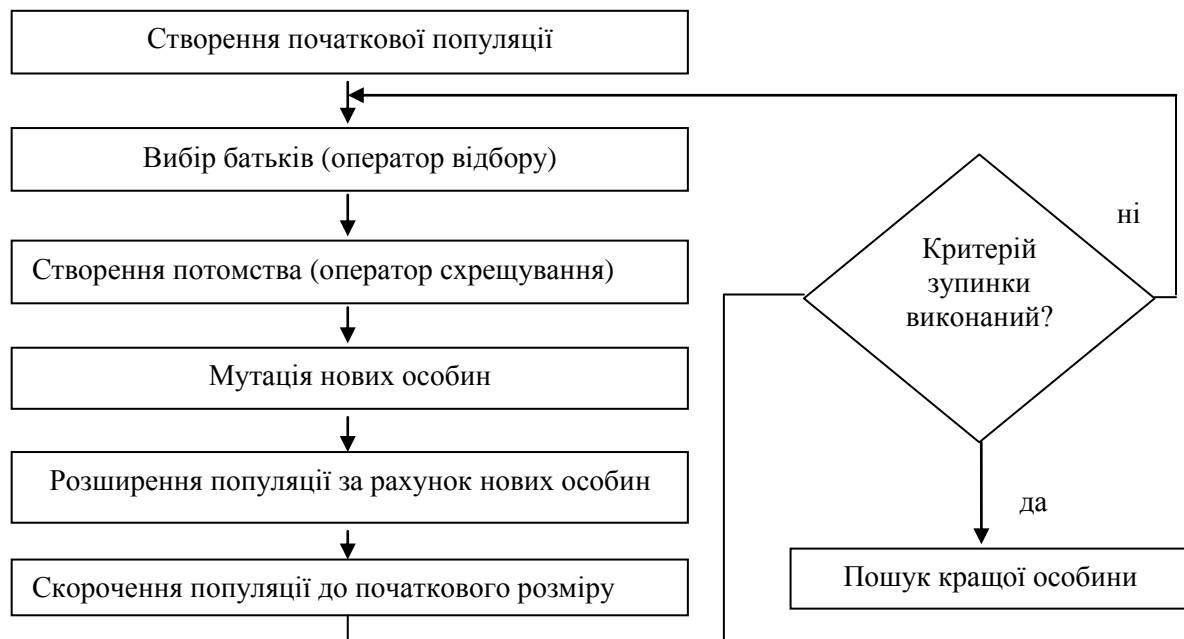


Рис.4.1. Схема роботи генетичного алгоритму

В якості критеріїв зупинки виконання алгоритму можуть використовуватися такі:

- сформовано задане число поколінь;
- популяція досягла заданої якості;
- досягнутий певний рівень збіжності.

Зазначимо, що останні два критерії пов'язані з заданою величиною придатності популяції або подібністю рядків у популяції.

Процес роботи ГА наведено на рис. 4.1.

Для ілюстрації основних понять і принципу дії ГА наведемо простий приклад на пошук максимуму.

Приклад 4.1. Знайти максимум функції $f(x) = x^2$ у діапазоні $0 < x < 31$. Тут в якості функції придатності виступає сама функція. Чим більше її значення, тим краще придатність хромосоми. Встановимо розмір популяції, рівний чотирьом рядкам. Початкова популяція та оцінка придатності наведена в табл. 4.1.

Таблиця 4.1. Початкова популяція й оцінка придатності

№ рядка	Початкова популяція	x	$f(x)$	Відносна придатність, %
1	01101	13	169	14,4
2	11000	24	576	49,2
3	01000	8	64	5,5
4	10011	19	361	30,9
			1170	100

Так як функція придатності другого рядка краща, відбираємо дві копії другого рядка і залишаємо перший і четвертий рядки в батьківському пулі. Відбір партнерів виконуємо випадковим чином: партнером першого рядка служить другий, партнером четвертого – теж другий.

Таблиця 4.2. Батьківський пул і схрещування

№ рядка	Батьківський пул	Парний рядок	До схрещування	Після схрещування
1	01101	2	0110[1]	01100
2	11000	1	1100[0]	11001
3	11000	4	11[000]	11011
4	10011	2	10[011]	10000

Положення точок схрещування також випадкове і вибирається таким чином: для пари з першого і другого рядків точка схрещування – після четвертого біта; для пари з другого – четвертого рядків – після другого біта (табл. 4.2).

Друге покоління без мутації наведено в табл. 4.3.

Таблиця 4.3. Друге покоління

Рядок	x	$f(x)$	Відносна придатність, %
01100	12	144	8,2
11001	25	625	35,6
11011	27	729	41,2
10000	16	256	14,7
		1754	100%

З табл. 4.3 видно, що третій рядок є кращим у другому поколінні і при значення $x = 27$ досить близько до шуканого максимуму. Очевидно, що через кілька кроків оптимальне рішення буде знайдено навіть без використання оператора мутації.

Кодування в генетичних алгоритмах

З біології відомо, що будь-який організм може бути представлений своїм фенотипом, який фактично визначає, чим є об'єкт в реальному світі (у зовнішньому середовищі). Генотип містить всю інформацію про об'єкт на рівні хромосомного набору. Для вирішення задачі необхідно представити кожен ознаку об'єкта у формі, придатній для використання в ГА. Кодування рішення задачі в хромосомі є ключовою проблемою застосування ГА. Проблема досліджена з різних сторін, наприклад, за характером відображення простору генотипів у простір фенотипів, коли окремі особини декодуються в рішення, та за властивостями перетворення, коли хромосоми регулюються за допомогою операторів.

У класичній роботі Д. Холланда кодування виконується з використанням бінарних рядків. Таке кодування для задач оптимізації має кілька недоліків унаслідок існування хеммінгова зсуву для пари закодованих значень, що мають велику *хеммінгову відстань* (число позицій, у яких відповідні символи двох слів однакової довжини різні), у той час, як ці величини належать до точок з мінімальною відстанню в фенотиповому просторі. Наприклад, пара 0111111111 і 1000000000 належить сусідніх точках у фенотипичному просторі (точки з мінімальною *евклідовою відстанню* – відстань між двома точками евклідового простору, обчислювана за теоремою Піфагора), але мають максимальну хеммінгову відстань у генотиповому просторі. Для подолання хеммінгова зсуву всі біти повинні змінюватися одночасно, але ймовірність такої події дуже мала.

Для багатьох задач, що зустрічаються в сучасних проблемах, досить важко уявити їх вирішення за допомогою бінарного кодування. Протягом останніх десяти років були розроблені різні методи кодування для забезпечення ефективного виконання ГА, які можуть бути розділені на такі класи:

- бінарне кодування;
- кодування дійсними числами;
- цілочисельне кодування;
- кодування загальної структури даних.

Більш докладно розглянемо бінарне кодування, яке використовується в класичному підході Д. Холланда. Нехай вектор допустимого рішення $x \in D$, де D – область пошуку рішень. Кожен компонент вектора $\bar{x} = (x_1, x_2, \dots, x_n)$ можна закодувати за допомогою цілого ненегативного числа $\beta_i \in [0, K_i], i = 1, n; (K_i - \text{число можливих дискретних значень } i\text{-ї змінної})$.

Введемо двійковий алфавіт $B_2 = \{0;1\}$. Для подання цілочисельного вектора $\beta = (\beta_1, \dots, \beta_n)$ в алфавіті B_2 необхідно визначити максимальне число двійкових символів θ , що достатньо для відо-

браження в двійковому коді будь-якого значення β_i з області її допустимих значень $[0, K_i]$. Незавжно бачити, що параметр θ повинен задовольняти нерівності

$$K < 2^\theta, (4.1)$$

де $K = \max K_i, \quad 1 \leq i \leq n$.

Тоді для $\beta_i (0 \leq \beta_i \leq 2^\theta)$ можна записати

$$\beta_i = \sum_{i=1}^n \alpha_i \cdot 2^{\theta-1}, (4.2)$$

де α_i – двійкове число (0 або 1); θ – довжина двійкового слова, що кодує ціле число β_i .

Приклад 4.2. Представимо у вигляді хромосомного рядка кількість $\beta_i = 19$.

З урахуванням формул (4.1) і (4.2) маємо: $\theta = 5$, так як $2^\theta = 2^5 = 32$.

За формулою (4.2) отримаємо:

$$\begin{aligned} 19 &= 1 \cdot 2^{5-1} + 0 \cdot 2^{5-2} + 0 \cdot 2^{5-3} + 1 \cdot 2^{5-4} + 1 \cdot 2^{5-5} = \\ &= 1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0. \end{aligned}$$

Рядок буде мати такий вигляд:

10011

В якості гена, тобто одиниці спадкового матеріалу, відповідального за формування ознак особини, приймемо бінарну комбінацію $e_0(\beta_i)$, яка визначає фіксоване значення цілочисельного коду b_i керованої змінної x_i у звичайному двійковому коді. В цьому випадку $e_0(\beta_i)$ має вигляд:

α_1	α_2	...	α_θ
------------	------------	-----	-----------------

θ

Одна особина α_k^t буде характеризуватися n генами, кожен з яких відповідає за формування цілочисельного коду відповідної керованої змінної. Тоді хромосому можна визначити в такому вигляді $E(X)$:

$\alpha_i^1 \dots \alpha_\theta^1$	$\alpha_i^2 \dots \alpha_\theta^2$	$\dots \dots \dots$	$\alpha_i^n \dots \alpha_\theta^n$
$e_\theta(\beta_1)$	$e_\theta(\beta_2)$	\dots	$e_\theta(\beta_n)$
<i>ген1</i>	<i>ген2</i>	\dots	<i>генn</i>
Локус 1	Локус 2	\dots	Локус n
ХРОМОСОМА			

Розташування певного гена в хромосомі називається локусом, а альтернативні форми одного і того ж гена, розташовані в одних і тих же локусах, називаються алелями.

Хромосома, що містить в своїх локусах конкретні значення алелей, визначає генотип (генетичний код) $E(\alpha_k^t)$, який містить всю спадкову генетичну інформацію про особини α_k^t , одержувану від предків і потім передається нащадкам. Кінцева множина всіх допустимих генотипів утворює генофонд.

Таким чином, у реалізації ГА хромосома являє собою бітовий рядок фіксованої довжини. При цьому кожній ділянці рядка відповідає ген. Довжина гена усередині рядка може бути однаковою або різною. Наприклад, для об'єкта з п'яти ознак, кожен з яких закодований геном довжиною в чотири елемента, загальна довжина хромосоми складе $5 \times 4 = 20$ бітів:

0010 1010 1001 0100 1101

При використанні N бітів для бінарної рядка (хромосоми) перетворення від двійкового коду рядка до десяткового значення здійснюється за формулою:

$$x_i = a_i + \text{decimal}(100\dots 010_2) \cdot \frac{b_i - a_i}{2^N - 1}$$

де a_i, b_i – нижня і верхня межі зміни i -ої ознаки; $\text{decimal}(100\dots 010_2)$ – десяткове значення бінарного рядка.

Кодування дійсними числами є кращим при вирішенні задач оптимізації функцій. Так як топологічна структура простору генотипів для цього виду кодування ідентична структурі у просторі фенотипів, то легко сформувані ефективні генетичні оператори запозиченням корисних прийомів у традиційних методів.

Цілочисельне кодування краще всього підходить для комбінаторних оптимізаційних задач.

У відповідності до кодування загальної структури даних, методи кодування можуть бути розділені на два види:

- одновимірний;
- багатовимірний.

У більшості випадків використовується одномірне кодування, однак багато проблем потребують вирішення у багатовимірному просторі, тому природно застосовувати метод багатовимірного кодування в таких ситуаціях.

Генетичні алгоритми виконуються на двох типах просторів: кодування і рішення, іншими словами, просторах генотипу та фенотипу. Генетичні оператори працюють у просторі генотипів, а оцінка і відбір відбуваються в фенотиповому просторі. Природний відбір – це зв'язок між хромосомами і поведінкою декодованих рішень.

4.3. Генетичні оператори

Визначено три основних види генетичних операторів:

- селекція;
- схрещування;
- мутація.

Перед тим як безпосередньо перейти до вивчення цих операторів, зупинимося докладніше на концепції пошуку та його застосування в ГА.

Пошук – один з універсальних методів вирішення завдань, у яких не можна заздалегідь визначити послідовність кроків, що приводять до бажаного результату. Зазвичай розглядаються два види пошуку:

- випадковий;
- локальний.

Випадковий пошук досліджує рішення цілком і здатний уникнути попадання в локальний оптимум. Локальний пошук використовує найкраще рішення і здатний піднятися вгору до локального оптимуму.

Ідеальний пошук повинен поєднувати обидва типи пошуку одночасно. Але практично такий пошук неможливо здійснити традиційними методами. ГА є класом загальноцільових пошукових методів, що комбінують елементи спрямованого і стохастичного типів пошуку, що призводить до хорошого балансу між дослідженням і використанням пошукового простору.

У традиційних ГА оператор схрещування застосовується як основний, і поведінка генетичної системи значною мірою залежить від нього. Оператор мутації, який здійснює спонтанні випадкові зміни в різних хромосомах, частково виконує роль фону. По суті, генетичні оператори проводять випадковий пошук і не можуть гарантувати появи покращеного потомства. Було виконано багато емпіричних досліджень порівняння мутації та схрещування, в результаті чого з'ясувалося, що іноді мутація грає більш важливу роль, ніж схрещування.

Є дві гіпотези пояснення того, як ГА використовують розподілену інформацію для формування хороших рішень:

- гіпотеза будівельних блоків;
- гіпотеза зміни керованої збіжності.

Згідно з першою гіпотезою схрещування комбінує ознаки від двох батьків для виробництва потомства. Іноді схрещування використовує кращі ознаки від обох батьків, формуючи дуже гарне потомство. Так як придатність особини часто залежить від складної взаємодії простих ознак, то важливо, щоб оператор був в змозі передати нащадкам ті набори ознак, які вносили вклад в оцінку придатності батьків.

Друга гіпотеза припускає використання збіжності популяції для обмеження пошуку. При збіжності популяції варіація придатності стає більш помітною. У той час як перша гіпотеза посилює комбінацію ознак, які вижили у батьківської популяції, друга гіпотеза посилює випадкову вибірку з розподілу, який є функцією поточної популяції.

Селекція

Основним принципом генетичних алгоритмів, по суті, є дарвінівський природний відбір. Селекція забезпечує рушійну силу ГА. Чим більше сила, тим швидше може завершитися генетичний пошук; при меншій величині цієї сили еволюційний процес буде повільніше, ніж необхідно.

Селекція направляє генетичний пошук в «обіцяючі» райони пошукового простору. Основний компонент ГА – це метод, використовуваний для переходу від однієї генерації до наступної. Існує багато можливих варіацій щодо вибору потенційних батьків і способи їх комбінування для відтворення потомства. Селекція може змінюватися від дуже трудомістких підходів, які потребують значних зусиль, до дуже легких схем. Протягом останніх 20 років було запропоновано багато методів, серед яких найбільше поширення одержали такі [4,6]:

Селекція за допомогою пропорційної рулетки. Метод рулетки – найвідоміший спосіб відбору. Основна ідея цього підходу полягає у визначенні ймовірності відбору (ймовірності виживання) для кожної хромосоми пропорційно величині її придатності. Процес відбору заснований на обертанні колеса рулетки стільки

разів, який розмір популяції, щоразу обираючи одну хромосому для нової популяції. Колесо служить методом відбору як стохастична вибіркова процедура.

В якості дуже спрощеного прикладу розглянемо популяцію з 10 індивідумів. Нехай ця популяція містить шість ідентичних хромосом, які позначимо як тип А і чотири ідентичних хромосоми типу В. Припустимо, що тип А має величину придатності, що дорівнює 12, а тип В – 7. Загальна придатність складе величину:

$$6 \times 12 + 4 \times 7 = 100.$$

Тип А займає 72% загальної придатності, а тип В – 28%. При таких вхідних даних «конструємо» рулетку, 72% площі якої позначимо міткою А, а 28% – міткою В. Для вибору наступної популяції даємо рулетці тільки 10 обертів (такий розмір популяції) і створюємо нову комбінацію з типів А і В. У створеній новій генерації буде міститися, в середньому, близько семи (заснованому на 72%) членів типу А і трьох (заснованому на 28%) членів типу В. Оскільки обертання колеса рулетки і момент його зупинки випадкові (в комп'ютерній реалізації використовується випадковий цифровий генератор), то дійсний поділ між А і В може бути і 0/10, 10/0, однак у середньому цей метод дає високі значення придатності.

Стохастична залишкова селекція. Розглянемо, як і вище, десятичленну популяцію типів А і В. Очікуване число копій типу А є 7,2, а для типу В – це число складе 2,8. У цьому методі селекції спочатку зосередимося на цілих частинах очікуваних величин: 7 і 2. Сім копій типу А розміщуються в наступній генерації так само, як і дві копії типу В. Для визначення ідентичності 10-го члена залишки використовуються для «зважування» колеса рулетки. Тут В отримує 80% всієї ваги, а тип А – тільки 20%. Цей метод буде давати внесок в нову генерацію у співвідношенні 7/3 або 8/2 між типами А і В.

Стохастична універсальна селекція. Цей метод – ще одна варіація попередніх двох, але вимагає трохи більшої уваги для йо-

го реалізації. Уявімо те ж саме колесо рулетки, 72% площі якої зазначено як А, а 28% – як В. Доповнимо це колесо зовнішнім кільцем з 10 рівних інтервалів (маркерів). Колесо рулетки робить один оборот. Для визначення складу наступної генерації розглядаються 10 маркерів: ті, які знаходяться навпроти площі А, позначаються як тип А; те ж саме відноситься і до В. Цей метод буде давати результати, схожі на попередній випадок: поділ у цьому випадку складе або $7/3$, або $8/2$.

Турнірна селекція. Турнірна селекція являє собою приклад процедури, що поєднує одночасно випадковий і детермінований підходи. Турніри схожі на маленькі «бої» між членами популяції, щоб побачити, хто візьме участь у наступному поколінні. Тут випадково визначається множина хромосом, з якої відбираються найкращі особини для репродукції. Число хромосом у цій множині називається розміром турніру, який найчастіше приймається рівним 2 (бінарний турнір). Після отримання пари хромосоми з більш високою оцінкою придатності впроваджуються в нову популяцію. Процес продовжується до того часу, поки не заповниться вся популяція.

Ранжирувана селекція. Ця процедура починається з ранжирування популяції за величиною придатності. Далі вводиться функція присвоювання, яка дає кожній хромосомі ймовірність включення в наступну генерацію. Хромосоми з більш високим рангом мають велику ймовірність включення. Функція присвоювання може бути лінійною або нелінійною (частіше – остання). Колесо рулетки будується зі слотами, що визначаються функцією присвоювання. Наступна генерація n розмірної популяції визначається після n обертів колеса. Ця процедура просуває селекцію до членів популяції, що володіють кращими характеристиками.

З перерахованих методів селекції найбільше поширення отримав метод пропорційної рулетки, однак потрібно вказати деякі недоліки процедури пропорційного відбору. У цій схемі на ранніх поколіннях може проявитися тенденція домінування «ду-

же хороших» особин у процесі відбору; на більш пізніх генераціях конкуренція серед таких хромосом стає менш сильною, і більшою мірою домінує випадковий пошук.

Крім зазначених способів селекції, існують так звані елітні методи, які гарантують, що в процесі селекції будуть зберігатися кращі (у сенсі придатності) члени популяції.

Найбільш часто використовується процедура збереження однієї найкращої хромосоми, якщо вона не пройшла, як інші, через селекцію, схрещування і мутацію. Елітний метод може бути застосований до будь-якої із зазначених вище схем селекції.

Схрещування

Схрещування – це крок, який реально підсилює генетичні алгоритми. Воно дозволяє розширити пошук у різних напрямках, оцінюючи привабливі рішення і дозволяючи двом рядкам «злучатися». Такий підхід може призвести до нащадків, які виявляться більш привабливим, ніж батьки.

Нехай популяція $P^t = (a_1^t, \dots, a_v^t)$ являє собою репродукційну групу, тобто сукупність v особин, будь-які дві з яких a_k^t , $a_l^t \in P^t$, $k \neq l$ можуть розмножуватися, виступаючи в ролі батьків (a_k^t – мати; a_l^t – батько). Тут під розмноженням мається на увазі властивість особин відтворювати одного або декількох собі подібних безпосередніх нащадків b_i^t , $i \geq 1$ і забезпечувати у них безперервність і спадкову спадкоємність якісних ознак батьків. Таким чином, цей чинник еволюційного розвитку популяції призводить до отримання нової генетичної інформації, яка містить різні комбінації алельних форм генів батьківських генотипів.

Розглянемо механізм розмноження двох батьківських особин $a_k^t, a_l^t \in P$ шляхом сигнамії (запліднення) їх репродуктивних клітин: материнської гамет (яйцеклітини) $E(a_k^t)$ і батьківської гамет (сперматозоїда) $E(a_l^t)$. В процесі сигнамії утворюється

батьківська зигота (запліднена клітина), здатна розвиватися в нову особину з передачею спадкових ознак (генетичною інформацією) від батьків їх нащадкам. Зигота містить одну пару з двох хромосом, які не відрізняються одна від іншої, які відбуваються від «батьківських» гамет: одна – від материнської гамети, а інша – від батьківської гамети. Такі хромосоми називаються гомологічними хромосомами.

В результаті акту сигнації алелі батьківських гамет можуть помінятися місцями в алельних генах гомологічної хромосоми, що дозволяє розглядати такі ситуації утворення зигот:

- ген з батьківської хромосоми переходить у материнську хромосому;
- ген з материнської хромосоми переходить у батьківську хромосому;
- відбувається взаємний обмін генами між материнською і батьківською хромосомами;
- батьківська і материнська хромосоми залишаються без зміни.

Таким чином, при утворенні зигот відбувається незалежна і випадкова розбіжність батьківських генів за алельними генами гомологічних хромосом зиготи незалежно від того, у якій із батьківських гамет вони були присутні до запліднення.

Заключним етапом розмноження особин є акт мейозу, під яким розуміється процес утворення гамет з батьківської зиготи шляхом незалежного розходження гомологічних хромосом дочірнім гаметам, що відтворюють потомство.

Процес розмноження двох особин повинен задовольняти першому і другому законам спадковості Менделя, які формулюються таким чином:

- два гени, що визначають ту чи іншу ознаку, не зливаються і не розчиняються один в іншому, але залишаються незалежними один від одного, розщеплюючись при формуванні гамет;

– батьківські гени, що визначають різні ознаки, які успадковуються незалежно один від одного.

Відповідно до першого закону гени (або відповідні їм ознаки батьків), що мають однакові алелі, зберігають свої значення в потомстві, тобто передаються з імовірністю, рівною 1, нащадку у спадок. Гени батьків, які мають різні алелі, передаються нащадку у спадок з імовірністю, що дорівнює 0,5, тобто половина гамет виявляється носієм алелі батька, а інша половина – алелі матері.

Відповідно до другого закону рекомбінація (обмін) генів в акті сигнамії може відбуватися або в якомусь одному алельному гені, або в декількох алельних генах одночасно, тобто передача алелей від батьків потомству може відбуватися у кожному алельному гені незалежно один від одного. При цьому може виявитися, що гамети нащадків або збігаються з батьківськими гаметами, або відрізняються від них в одному або декількох локусах.

Алгоритм схрещування може бути описаний такою послідовністю кроків.

1. З популяції випадково вибираються двоє потенційних батьків.

2. Виконується жеребкування (комп'ютерне) для визначення – проводити чи не проводити схрещування.

3. При позитивному варіанті відповіді на попередньому кроці випадково вибирається точка розщеплення.

4. Кожний вихідний батьківський рядок перерізується в точці розщеплення.

5. Проводиться обмін генетичним матеріалом ліворуч і праворуч від точки розщеплення і з'єднання отриманих рядків для формування потомства.

Схема схрещування показана на рис.4.2.

Наведена схема називається *одноточковим схрещуванням*, так як є тільки один розріз у кожній хромосомі.

Батьки	
010011000	1010
110010101	1100
Нащадки	
010011000	1100
110010101	1010

Рис.4.2. Схема схрещування

Потрібно зазначити, що процедура схрещування є досить критичною для пошуку найкращих рішень, оскільки хороші з точки зору придатності рядки можуть бути сильно спотворені. У разі, якщо схрещування відбувається досить часто, то пошук – хаотичний. і неадекватно отримує користь з попередніх успіхів. Рідкісні (нечасті) схрещування утримують пошук "у вузьких областях простору пошуку".

Поряд з одноточковим є *двоточкове схрещування*, яке включає два розрізи і може бути зображено так, як показано на рис. 4.3.

Батьки		
011	001	1
110	111	1
Нащадки		
011	111	1
110	001	1

Рис.4.3. Схема двоточкового схрещування

Для усунення небажаних ситуацій були розроблені схеми багатоточкового схрещування, зокрема рівномірного. Останнім

потенційно дозволяється будь-яким чином здійснювати обмін і сформувати потомство. Тут завдання вирішується на побітовій основі (bitby"bitbasis) для з'ясування того, як рядки потомства співвідносяться з батьківськими рядками.

В якості прикладу розглянемо два п'ятибітових рядка. Для створення схрещеного образу скористаємося для кожної бітової позиції жеребом (монета), замість орла і решки якого визначимо два результати: «same – такий самий» та «swar "обмін». Результат показаний у табл. 4.4.

Далі отриманий образ «same"swar"same"swar"same» використовуємо для формування двох нащадків. Для першої бітової позиції образ «same»; батько 1 тутмає 0, тому нащадок 1 отримує також 0; батько 2 – 1, отже, нащадок 2 отримує 1. Для другої бітової позиції образ «swar», тому нащадок 1 отримує 1 від батьків1, а нащадок 2 – 0 від батьків 2. Остаточний результат наведено в останніх двох рядках табл. 4.4.

Таблиця 4.4. Рівномірне схрещування

Рядки	Положення біта				
	1	2	3	4	5
Батько 1	0	0	0	1	1
Батько 2	1	1	1	0	0
Схрещений образ	same	swar	same	swar	same
Нашадок 1	0	1	0	0	1
Нашадок 2	1	0	1	1	0

Рівномірне схрещування дає більшу гнучкість при комбінуванні рядків, що є привабливою властивістю при роботі з ГА. Підводячи підсумок розгляду оператора схрещування, зазначимо, що схрещуванню підлягає не вся популяція рядків, а тільки її частина, яка визначається ймовірністю схрещування (наприклад, $p_{\text{скр.}} = 0,6$ означає, що лише 60% рядків з популяції будуть утворювати батьківські пари).

Мутація

При схрещуванні генотипи нащадків містять нові сполучення алельних форм генів батьків, що ведуть до нових кількісних ознак нащадків (ступенів придатності). Проте генетична інформація, що міститься в хромосомному наборі батьків і нащадків, не змінюється, так як в результаті розмноження особин шляхом сигнамії і мейозу частоти алелей залишаються постійними, а змінюються тільки частоти генотипів. Джерелом генетичної мінливості особин є мутація, під якою розуміється зміна якісних ознак особин у результаті появи нових алельних форм в окремих генах або цілком у хромосомі. Тим самим у кожному поколінні мутації постачають у хромосомний набір популяції безліч різних генетичних варіацій, притаманних особинам, яких у подальшому будемо називати мутантами.

Процес зміни складу генів у хромосомі шляхом мутації називається мутагенезом. Найголовніше полягає в тому, що цей фактор еволюції популяції є джерелом нової генетичної інформації, яка не містилася раніше в генах батьків та їх нащадків. Мутації відбуваються незалежно від того, чи приносять вони особині шкоду або користь. Вони не спрямовані на підвищення або зниження ступеня пристосованості особини, а тільки справляють структурні зміни в алельних формах генів, що призводить до зміни кількісних ознак особини. В принципі, комбінація мутацій може привести до виникнення нових форм алелей в деяких генах генотипу мутанта, які забезпечують збільшення його ступеня пристосованості до зовнішнього середовища.

Найбільш простим видом мутацій є точкова мутація, пов'язана із зміною алелі батьківського гена в одному з бітовогенної інформації (0 замінюється на 1 або навпаки), наприклад, у рядку виду 0 1 1 0 0 замінюється середній елемент, в результаті чого отримуємо такий рядок: 0 1 0 0 0. Крім того, мутація може бути інверсійною, при якій відбувається зміна всіх компонентів рядка, наприклад, рядок 0 1 1 0 0 замінюється рядком 1 0 0 1 1.

Оператор мутації вводить елемент варіабельності в популяцію, так як різко змінює деякі рішення. Основним параметром мутації є її імовірність, яку задає користувач. Зазвичай її величина досить мала (близько 0,01 і нижче), щоб, з одного боку, розширити область пошуку, а з іншого – не привести до таких змін нащадків, які будуть далекі від прийнятних рішень.

Приклад 4.3. Розглянемо роботу ГА в задачі мінімізації функції чотирьох змінних

$$J(x_1, x_2, x_3, x_4) = (x_1 - 5)^2 + (x_2 - 6)^2 + (x_3 - 7)^2 + (x_4 - 8)^2,$$

мінімум, якій досягається при $x_1 = 5, x_2 = 6, x_3 = 7, x_4 = 8$. Задамо межі зміни змінних: $1 \leq x_j \leq 10$ для всіх $j = 1, 2, 3, 4$, і визначимо основні поняття ГА.

Особина, або хромосома P^i – це сукупність значень змінних, званих генами, $P^i = (x_1^i, x_2^i, x_3^i, x_4^i), i = 0, 1, 2, \dots, n$. Наприклад, нехай нульова особина P^0 містить такі значення генів:

$$P^0 = (x_1^0, x_2^0, x_3^0, x_4^0) = (1, 1, 1, 1).$$

Значення критерію буде дорівнює

$$J(P^0) = J(1, 1, 1, 1) = (1 - 5)^2 + (1 - 6)^2 + (1 - 7)^2 + (1 - 8)^2 = 126.$$

Популяція $\Pi = (P^1, P^2, \dots, P^n)$ містить кілька особин P^1, P^2, \dots, P^n , кількість яких n обмежено умовою $n \leq 5$.

Ген j характеризується величиною x_j і положенням j в особині. Можна випадково вибрати j -ї ген:

$$j = 1 + (m - 1) \cdot Rnd, \quad (4.3)$$

і випадковим чином змінити його величину, тобто мутувати j -ї ген:

$$x_j = x_j^{\min} + (x_j^{\max} - x_j^{\min}) \cdot Rnd, \quad (4.4)$$

де m – номер останнього гена в особини; x_j^{\min}, x_j^{\max} – нижня і верхня границі зміни змінної x_j ; Rnd – генератор випадкових чисел, виробляє при кожному зверненні до нього число в інтервалі $[0,1]$.

У нашому випадку якщо $m = 4, x_i^{\min} = 1, x_j^{\max} = 10$ для всіх $j = 1, 2, 3, 4$, то при підстановці цих значень у формулу (4.3) і (4.4) отримаємо:

$$j = 1 + 3 \cdot Rnd \quad (4.5)$$

і

$$x_j = 1 + 9 \cdot Rnd \quad (4.6)$$

Для простоти будемо оперувати цілочисельними значеннями j і x_j . ГА складається з операторів створення початкової популяції, відбору, схрещування, мутації і редукції (обчислення значення критерію).

Вихідна популяція $\Pi^0 = (P^1, P^2, P^3)$ складається з трьох особин P^1, P^2, P^3 , одержуваних з нульовою особиною P^0 . Нехай при першому зверненні до формули (4.5) знайдено кількість генів $n=2$, що підлягають мутації. Виконуючи $n=2$ звернення до формули (4.5), визначаємо номери $j_1=1, j_2=3$, також виконуючи $n=2$ звернень до формули (4.6), визначаємо значення $x_1^1 = 5, x_3^1 = 6$ мutowаних генів. Замінюючи гени $x_1^0 = 1, x_3^0 = 1$ у нульової особини P^0 на відповідні мutowані $x_1^1 = 5, x_3^1 = 6$, отримуємо першу особину:

$$P^1 = (x_1^1, x_2^0, x_3^1, x_4^0) = (5, 1, 6, 1).$$

Аналогічним чином, при $n=1, j_1=4$ і $x_4^2 = 7$ отримуємо другу особину:

$$P^2 = (x_1^0, x_2^0, x_3^0, x_4^2) = (1, 1, 1, 7),$$

а при $n = 3, j_1 = 1, j_2 = 2, j_3 = 4$ і $x_1^3 = 2, x_2^3 = 3, x_4^3 = 4$ – третю особину

$$P^3 = (x_1^3, x_2^3, x_3^0, x_4^3) = (2, 3, 1, 4).$$

Оператор відбору обчислює функцію J для кожної особи:

$$J(P^1) = J(5, 1, 6, 1) = (5-5)^2 + (1-6)^2 + (6-7)^2 + (1-8)^2 = 75$$

$$J(P^2) = J(1, 1, 1, 7) = (1-5)^2 + (1-6)^2 + (1-7)^2 + (7-8)^2 = 78$$

$$J(P^3) = J(2, 3, 1, 4) = (2-5)^2 + (3-6)^2 + (1-7)^2 + (4-8)^2 = 70,$$

і вибирає дві особини - батьків P^1 і P^3 з мінімальними значеннями критеріїв $J(P^1) = 75, J(P^3) = 70$.

Оператор схрещування передає нащадкам властивості батьків. Спочатку за формулою (4.5) визначається номер гена $k=2$, після якого обмінюються місцями підрядки з утворенням нащадків – особин P^4 і P^5 , які успадковують властивості батьків

$$\begin{matrix} & k=2 \\ \text{Батьки} \left\{ \begin{array}{l} P^1 = (5, 1, 6, 1), \\ P^3 = (2, 3, 1, 4). \end{array} \right. & \text{Нашадки} \left\{ \begin{array}{l} P^4 = (5, 1, 1, 4), \\ P^5 = (2, 3, 6, 1). \end{array} \right. \end{matrix}$$

Оператор мутації при виконанні умови $Rnd \leq \varepsilon$, де $0 < \varepsilon < 1$, за формулою (4.5) визначає номер $\lambda = 2$ генів, які переставляються в нащадках - особинах P^4 і P^5 з утворенням мutowаних нащадків \bar{P}^4 і \bar{P}^5 :

$$\lambda = 2$$

$$\text{Нашадки} \left\{ \begin{array}{l} P^4 = (5, 1, 1, 4), \\ P^5 = (2, 3, 6, 1). \end{array} \right. \Rightarrow \left\{ \begin{array}{l} \bar{P}^4 = (5, 3, 1, 4), \\ \bar{P}^5 = (2, 1, 6, 1). \end{array} \right.$$

Оператор редуції обчислює значення критерію для нащадків:

$$J(\bar{P}^4) = J(5, 3, 1, 4) = (5-5)^2 + (3-6)^2 + (1-7)^2 + (4-8)^2 = 61$$

$$J(\bar{P}^5) = J(2, 1, 6, 1) = (2-5)^2 + (1-6)^2 + (6-7)^2 + (1-8)^2 = 84.$$

З п'яти особин

$$P^1 = (5, 1, 6, 1), P^2 = (1, 1, 1, 7), P^3 = (2, 3, 1, 4), \bar{P}^4 = (5, 3, 1, 4), \bar{P}^5 = (2, 1, 6, 1)$$

з критеріями

$$J(P^1) = 75, J(P^2) = 78, J(P^3) = 70, J(\bar{P}^4) = 61, J(\bar{P}^5) = 84$$

вибираються три особини P^1, P^3, \bar{P}^4 , які мають мінімальні значення критеріїв і утворюють нову вихідну популяцію

$$P^0 = (P^1, P^3, \bar{P}^4)$$

Слід зазначити, що значення критеріїв $J(P^1) = 75, J(P^3) = 70, J(\bar{P}^4) = 61$ для особин P^1, P^3, \bar{P}^4 набагато менше, ніж величина критерію $J(P^0) = 126$ для нульової особини P^0 . Отже, гени-змінні x_1, x_2, x_3, x_4 змінюються в напрямі мінімізації J . Наведені дії складають один повний цикл розрахунку ГА, званої епохою. ГА завершує роботу, якщо досягнута задана кількість епох або не змінюється величина критерію J , що свідчить про досягнення мінімуму. В іншому випадку, з новою вихідною популяцією повторюються дії операторів відбору, схрещування, мутації, редукції і т. ін.

4.4. Прийоми виконання генетичних алгоритмів

Розглянемо дію ГА на проблемі оптимізації функції багатьох змінних.

Нехай завдання полягає в максимізації функції k змінних $f(x_1, x_2, \dots, x_k)$; при цьому кожна змінна $x_i (i = \overline{1, k})$ приймає значення з області $D_i = [a_i, b_i]$ та $f(x_1, x_2, \dots, x_k) > 0$ для всіх $x_i \in D_i$.

Встановимо необхідну точність оптимізації функції f : два знаки після коми. Ясно, що кожна область D_i повинна бути розділена на $(b_i - a_i) \cdot 10^2$ рівних відрізків.

Позначимо через m_i найменше число, яке задовольняє нерівності

$$(b_i - a_i) \cdot 10^2 \leq 2^{m_i} - 1.$$

Тоді кожна змінна x_i кодується як бінарна рядок довжиною m_i , що задовольняє заданій точності.

Кожна хромосома (потенційне рішення) представляється бінарним рядком завдовжки $m = \sum_{i=1}^k m_i$. У цьому рядку перші m_1 бітів відображають x_1 з діапазону $[a_1, b_1]$, другі m_2 – з діапазону $[a_2, b_2]$ і т.ін.

У результаті хромосома може мати такий вигляд:

$$\underbrace{01010111}_{m_1} \underbrace{000111}_{m_2} \underbrace{100}_{m_k}$$

Крім того, задамо розмір популяції M (число хромосом). Далі робота ГА видається цілком очевидною відповідно до наведеного вище алгоритму:

– у кожній генерації оцінюється окрема хромосома на предмет її придатності з використанням функції f на декодованому наборі змінних:

– відбирається нова популяція з урахуванням розрахованої придатності;

– з допомогою операторів схрещування і мутації хромосоми комбінуються в нову популяцію.

Після деякого числа генерацій, коли не спостерігається поліпшення популяції, краща хромосома є оптимальним (можливо глобальним) рішенням. Часто алгоритм зупиняють після фіксованого числа ітерацій.

Розглянемо деякі кроки більш докладно.

Селекція. Для процесу селекції служить рулетка з розмірами секторів, пропорційних придатності кожного рядка. Розробка такої рулетки складається з таких кроків:

– обчислюється придатність $\mu(a_j)$ для кожної хромосоми:

$$a_j, j = \overline{1, M};$$

– знаходиться загальна функція придатності всієї популяції:

$$F = \sum_{j=1}^M \mu(a_j);$$

– визначається ймовірність вибору p_j для кожної хромосоми a_j :

$$p_j = \mu(a_j) / F;$$

– обчислюється кумулятивна (накопичена) ймовірність q_j для кожної хромосоми:

$$q_j = \sum_{j=1}^{j^*} p_j.$$

Процес селекції заснований на обертанні колеса M разів, і кожен раз відбирається одна хромосома в нову популяцію таким чином:

– генерується випадкове число r з діапазону $[0...1]$;

– якщо $r < q_1$, то вибирається перша хромосома a_1 ; в іншому випадку відбирається j -та хромосома a_j ($2 \leq j \leq M$) так, щоб $q_{j-1} < r \leq q_j$.

Очевидно, що деякі хромосоми будуть обрані більше, ніж один раз. Кращі хромосоми дають більше копій, середні – залишаються незмінними, погані – помирають. Нові рішення на цьому етапі не створюються.

Схрещування. Задається ймовірність схрещування p_c . Очікуване число хромосом, які піддаються схрещуванню, становить $p_c M$.

Для кожної хромосоми (нової) популяції:

– генерується випадкове число s з діапазону $[0...1]$;

– якщо $r < p_c$, то ця хромосома вибирається для схрещування.

Таким чином відбираються особини для схрещування. Вибір точки схрещування теж випадковий: генерується випадкове число s з діапазону $[1...(m-1)]$ (m – довжина хромосоми). Це число s визначає точку схрещування.

У результаті дві хромосоми $(b_1 b_2 \dots b_s b_{s+1} \dots b_m)$ і $(c_1 c_2 \dots c_s c_{s+1} \dots c_m)$ замінюються парою нащадків $(b_1 b_2 \dots b_s c_{s+1} \dots c_m)$ і $(c_1 c_2 \dots c_s b_{s+1} \dots b_m)$.

Мутація. Задається ймовірність мутації p_m . Очікуване число змінених біт складе $p_m \cdot m \cdot M$. Кожен біт у всіх хромосомах у всій популяції має рівний шанс зазнати мутації, тобто змінитися з 0 на 1 або навпаки. Це здійснюється таким чином:

– генерується випадкове число r з діапазону $[0...1]$;

– якщо $r < p_m$, то біт змінюється.

Після відбору, схрещування і мутації нова популяція готова для подальшого оцінювання. Отримані оцінки використовуються для побудови нової рулетки з секторами, пропорційними поточним значенням функції придатності. Інша частина еволюції представляє по суті циклічне повторення процесу.

Приклад 4.4. Скористаємося наведеними поясненнями для вирішення наступного завдання [2]. Знайти максимум функції:

$$f(x_1, x_2) = 21,5 + x_1 \sin(4\pi x_1) + x_2 \sin(20\pi x_2),$$

$$\text{де } -3,0 \leq x_1 \leq 12,1 \quad 4,1 \leq x_2 \leq 5,8.$$

Крім того, прийmemo розмір популяції $M = 20$; ймовірність схрещування $p_c = 0,25$; імовірності мутації $p_m = 0,01$. Припустимо, що

необхідна точність становить чотири цифри після коми для кожної змінної. Тоді діапазон для змінної x_1 , якої становить 15,1, повинен бути розділений на $15,1 \cdot 10000$ рівних відрізків. Це означає, що для першої частини хромосоми потрібно 18 бітів, так як $2^{17} \leq 15100 \leq 2^{18}$.

Для другої змінної x_2 з діапазоном рівним 1,7, умова встановленої точності вимагає, щоб весь діапазон був розділений на $1,7 \cdot 10000$ рівних відрізків. Таким чином, для цієї змінної необхідно 15 бітів, оскільки

$$2^{14} \leq 17000 \leq 2^{15}.$$

Загальна довжина хромосоми (вектор потенційного рішення) складе $m = 18 + 15 = 33$ бітів, з яких перші 18 кодують першу змінну, а решту 15 – другу. Розглянемо, наприклад, такий рядок:

(010001001011010000 111110010100010).

Перші 18 бітів визначають таке значення змінної x_1 :

$$\begin{aligned} x_1 &= -3,0 + \text{decimal}(010001001011010000_2) \cdot \frac{12,1 - (-3,0)}{2^{18} - 1} = \\ &= -3,0 + 70352 \cdot \frac{15,1}{262143} = -3,0 + 4,05 = 1,05. \end{aligned}$$

Решта 15 бітів, декодовані за аналогією з вищенаведеною рівністю, дають для змінної x_2 значення, що дорівнює 5,75. Таким чином, хромосома (010001001011010000 111110010100010) відповідає $(x_1, x_2) = (1,05; 5,75)$, що визначає для оптимізованої функції таке значення:

$$f(x_1, x_2) = f(1,05; 5,75) = 20,25.$$

Створимо початкову популяцію, що складається з 20 рядків, у кожному з яких значення 33 бітів ініціюються випадковим чином. Покладемо, що після ініціювання отримана популяція, наведена в табл. 4.5.

Таблиця 4.5. Початкова популяція

Номер рядка	Рядок-хромосома
1	100110100000001111111010011011111
2	111000100100110111001010100011010
3	000010000011001000001010111011101
4	100011000101101001111000001110010
5	000111011001010011010111111000101
6	000101000010010101001010111111011
7	001000100000110101111011011111011
8	100001100001110100010110101100111
9	010000000101100010110000001111100
10	000001111000110000011010000111011
11	011001111110110101100001101111000
12	110100010111101101000101010000000
13	111011111010001000110000001000110
14	010010011000001010100111100101001
15	111011101101110000100011111011110
16	110011110000011111100001101001011
17	011010111111001111010001101111101
18	011101000000001110100111110101101
19	000101010011111111110000110001100
20	101110010110011110011000101111110

Тепер необхідно декодувати кожен хромосома й обчислити функцію придатності кожного рядка. Ясно, що в цьому прикладі придатність визначається самою оптимізованою функцією. Після декодування отримуємо результат, показаний у табл. 4.6.

З отриманих даних видно, що друга хромосома володіє найменшою придатністю, а хромосома a_{15} – найбільшою.

Перейдемо до конструювання рулетки, необхідної для процесу селекції. Загальна придатність всієї популяції становить величину

$$F = \sum_{j=1}^{20} \mu(a_j) = 387,77.$$

Таблиця 4.6. Придатність початкової популяції

Номер рядка	Функція	Придатність
1	$f(6,08;5,65)$	26,01
2	$f(10,34;4,38)$	7,58
3	$f(-2,51;4,39)$	19,52
4	$f(5,27;5,59)$	17,40
5	$f(-1,25;4,73)$	25,34
6	$f(-1,81;4,39)$	18,10
7	$f(-0,99;5,68)$	16,02
8	$f(4,91;4,70)$	17,95
9	$f(0,79;5,38)$	16,12
10	$f(-2,55;4,79)$	21,27
11	$f(3,13;4,99)$	23,41
12	$f(9,35;4,23)$	15,01
13	$f(11,13;5,37)$	27,31
14	$f(1,33;5,15)$	19,87
15	$f(11,08;5,05)$	30,06
16	$f(9,21;4,99)$	23,86
17	$f(3,36;4,57)$	13,69
18	$f(3,84;5,15)$	15,41
19	$f(-1,74;5,39)$	20,09
20	$f(7,93;4,75)$	13,66

Ймовірності вибору p_j для кожної хромосоми відповідно до вищезазначеного правила наведено в табл. 4.7.

Таблиця 4.7. Значення ймовірності кожної хромосоми

Номер рядка	Ймовірність p_j	Номер рядка	Ймовірність p_j
1	0,067	11	0,060
2	0,019	12	0,038
3	0,050	13	0,070
4	0,044	14	0,051
5	0,065	15	0,077
6	0,046	16	0,061
7	0,041	17	0,035
8	0,046	18	0,039
9	0,041	19	0,051
10	0,054	20	0,035

Кумулятивні ймовірності для кожної хромосоми наведено в табл. 4.8.

Таблиця 4.8. Кумулятивні ймовірності кожної хромосоми

Номер рядка	Ймовірність q_j	Номер рядка	Ймовірність q_j
1	0,067	11	0,538
2	0,086	12	0,577
3	0,137	13	0,647
4	0,181	14	0,698
5	0,247	15	0,776
6	0,293	16	0,837
7	0,335	17	0,873
8	0,381	18	0,912
9	0,423	19	0,964
10	0,478	20	1,000

Далі необхідно зробити 20 обертів рулетки, щоразу забираючи єдину хромосому для нової популяції. Нехай випадкова послідовність 20 чисел з діапазону $[0...1]$ має вигляд, показаний в табл. 4.9.

Таблиця 4.9. Випадкові числа з діапазону $[0,1]$

Розіграні числа				
0,513	0,175	0,308	0,534	0,947
0,171	0,702	0,226	0,494	0,424
0,703	0,389	0,227	0,368	0,983
0,005	0,765	0,646	0,767	0,780

Перше число r_1 більше, ніж q_{10} і менше, ніж q_{11} , тому для нової популяції вибирається хромосома a_{11} ; друге число r_2 більше, ніж q_3 і менше, ніж q_4 . Отже, другий для нової популяції вибирається рядок a_4 і т.і.

Остаточна нова популяція має вигляд, наведений у табл. 4.10.

Як видно з табл. 4.10, найгірший у початковій популяції рядок 2 після селекції не потрапив у наступну генерацію, а най-

кращий у початковій популяції рядок 15 з'явився в новій популяції три рази.

Таблиця 4.10. Нова популяція хромосом

Номер рядка	Рядок-хромосома	Старий номер рядка
1	011001111110110101100001101111000	11
2	100011000101101001111000001110010	4
3	001000100000110101111011011111011	7
4	011001111110110101100001101111000	11
5	000101010011111111110000110001100	19
6	100011000101101001111000001110010	4
7	111011101101110000100011111011110	15
8	000111011001010011010111111000101	5
9	011001111110110101100001101111000	11
10	000010000011001000001010111011101	3
11	111011101101110000100011111011110	15
12	010000000101100010110000001111100	9
13	000101000010010101001010111111011	6
14	100001100001110100010110101100111	8
15	101110010110011110011000101111110	20
16	100110100000001111111010011011111	1
17	000001111000110000011010000111011	10
18	111011111010001000110000001000110	13
19	111011101101110000100011111011110	15
20	110011110000011111100001101001011	16

Наступним кроком у проведенні ГА є схрещування, яке застосовується до отриманої нової популяції. Задана ймовірність схрещування становить величину $p_c = 0,25$, тому в середньому має зазнати схрещування 25% вихідних хромосом. Тут робимо таким чином: для кожної хромосоми в новій популяції генеруємо випадкове число r з діапазону $[0...1]$; якщо $r < 0,25$, то вибираємо дану хромосому для схрещування.

Припустимо, що послідовність випадкових чисел діапазону $[0...1]$ вийшла така, як показано в табл. 4.11.

Таблиця 4.11. Випадкові числа з діапазону [0,1]

Розіграні числа				
0,82	0,15	0,62	0,31	0,34
0,91	0,51	0,40	0,60	0,78
0,03	0,86	0,16	0,67	0,75
0,58	0,38	0,20	0,35	0,82

З табл. 4.11 випливає, що для схрещування відбираються хромосоми з номерами 2, 11, 13 і 18, оскільки значення випадкових чисел на цих позиціях менше ніж 0,25.

Зазначимо, що в даному випадку число відібраних хромосом вийшло парним, тому легко скласти батьківські пари. В іншому випадку необхідно додати або прибрати одну хромосому. Склад батьківських пар також випадковий, наприклад, в якості однієї такої пари виберемо рядки a_2 , a_{11} і інший – рядки a_{13} , a_{18} . Для кожної з цих двох пар генеруємо випадкове число s з діапазону $[1...32]$ (нагадаємо, що 32 – загальне число бітів у хромосомі), що визначає положення точки схрещування. Для першої пари це число складе 9, а для другої – 20.

Перша пара хромосом

$$a_2 = 100011000 \mid 101101001111000001110010$$

$$a_{11} = 111011101 \mid 101110000100011111011110$$

після схрещування дає таку пару нащадків:

$$a_2^* = 100011000 \mid 101110000100011111011110$$

$$a_{11}^* = 111011101 \mid 101101001111000001110010.$$

Друга пара хромосом

$$a_{13} = 00010100001001010100 \mid 1010111111011$$

$$a_{18} = 11101111101000100011 \mid 0000001000110$$

у результаті схрещування дає таку пару нащадків:

$$a_{13}^* = 00010100001001010100 \mid 0000001000110$$

$$a_{18} = 11101111101000100011 \mid 1010111111011.$$

Після схрещування популяція хромосом набуває вигляду, що показаний у табл. 4.12.

Перейдемо тепер до оператора мутації, який виконується на побітовій основі. Задана ймовірність мутації $p_m=0,01$, тому очікуване число мутованих біт складе 1% від загального числа бітів у популяції. В останній є $33 \cdot 20 = 660$ бітів. Отже, у середньому число бітів-мутантів складе 6–7 одиниць.

Таблиця 4.12. Популяція хромосом після схрещування

Номер рядка	Рядок-хромосома
1	011001111110110101100001101111000
2*	100011000101110000100011111011110
3	001000100000110101111011011111011
4	011001111110110101100001101111000
5	000101010011111111110000110001100
6	100011000101101001111000001110010
7	111011101101110000100011111011110
8	000111011001010011010111111000101
9	011001111110110101100001101111000
10	000010000011001000001010111011101
11*	111011101101101001111000001110010
12	010000000101100010110000001111100
13*	000101000010010101000000001000110
14	100001100001110100010110101100111
15	101110010110011110011000101111110
16	100110100000001111111010011011111
17	000001111000110000011010000111011
18*	111011111010001000111010111111011
19	111011101101110000100011111011110
20	110011110000011111100001101001011

Примітка. Схрещені хромосоми відмічені зірочкою.

Кожний біт у популяції має рівний шанс зазнати мутації, тому для кожного біта генеруємо випадкове число r з діапазону $[0...1]$; якщо $r < 0,01$, то даний біт мутує. В цілому, необхідно розіграти 660 випадкових чисел, з яких у даному випадку тільки п'ять задовольняють необхідним умовам. Положення біта і відповідне значення випадкового числа наведено в табл. 4.13.

Таблиця 4.13. Позиція мутованого біта в популяції

Позиція біта	Випадкове число
112	0,00021
349	0,00994
418	0,00880
429	0,00542
602	0,00283

Для визначення положення мутованого біта в рядках популяції скористаємося табл. 4.14.

Таблиця 4.14. Положення мутованого біта в рядках популяції

Позиція біта	Номер хромосоми	Номер біта в хромосомі
112	4	13
349	11	19
418	13	22
429	13	33
602	19	8

З табл. 4.14 видно, що чотири хромосоми піддалися мутації, причому одна з рядків з номером 13 двічі змінила значення бітів. Остаточна популяція після операторів схрещування і мутації наведена в табл. 4.15.

У цій же таблиці в останньому стовпці наведені значення функції придатності, отримані для вихідної популяції після селекції, схрещування і мутації. Кращий рядок має значення функції придатності, рівне 33,35, що перевищує найбільшу величину вихідної популяції. Крім того, і загальна придатність, рівна 447,04, набагато перевищує аналогічну величину на початку роботи ГА. Таким чином, за один крок процедури виконання ГА вдалося значно просунути вперед на шляху пошуку максимального значення даної функції. Далі необхідно знову застосувати селекцію, схрещування і мутації, оцінити отриману генерацію з точки зору її придатності і т. ін. до того часу, поки не буде задовольняти утвореним умовам зупинки.

Таблиця 4.15. Популяція після схрещування і мутації

Номер рядка	Хромосома	Функція придатності
2 [*]	100011000101110000100011111011110	$f(5,27;5,05)= 18,20$
4 ^{**}	0110011111100 0 10101100001101111000	$f(3,13;4,99)= 23,41$
11 ^{**}	111011101101101001 0 11000001110010	$f(11,08;4,74)= 33,35$
13 ^{**}	000101000010010101000 1 00001000111	$f(-1,81;4,20)= 22,69$
18 [*]	111011111010001000111010111111011	$f(11,13;5,66)= 27,59$
19 ^{**}	1110111 1 1101110000100011111011110	$f(11,05;5,05)= 27,60$
1	011001111110110101100001101111000	$f(3,13;4,99)= 23,41$
3	001000100000110101111011011111011	$f(-0,99;5,68)= 16,02$
5	000101010011111111110000110001100	$f(-1,74;5,39)= 20,09$
6	100011000101101001111000001110010	$f(5,27;5,59)= 17,40$
7	111011101101110000100011111011110	$f(1,08;5,05)= 30,06$
8	000111011001010011010111111000101	$f(-1,25;4,73)= 25,34$
9	011001111110110101100001101111000	$f(3,13;4,99)= 23,41$
10	000010000011001000001010111011101	$f(-2,51;4,39)= 19,52$
12	010000000101100010110000001111100	$f(0,79;5,38)= 16,12$
14	100001100001110100010110101100111	$f(4,91;4,70)= 17,95$
15	101110010110011110011000101111110	$f(7,93;4,75)= 13,66$
16	100110100000001111111010011011111	$f(6,08;5,65)= 26,01$
17	000001111000110000011010000111011	$f(-2,55;4,73)= 21,27$
20	110011110000011111100001101001011	$f(9,21;4,99)= 23,86$

Примітка. Мутовані рядки позначені двома зірочками, а змінені біти виділені жирним шрифтом.

4.5. Програмне забезпечення генетичних алгоритмів

Одна з переваг використання генетичних алгоритмів для вирішення різних завдань полягає в тому, що для реалізації ГА не потрібно створювати окремий програмний продукт. Єдине, що потрібно від користувача – це уявити шукане рішення у вигляді хромосоми і сформулювати функцію придатності. Потім реалізація ГА відбувається незалежно від конкретики розглянутої задачі. Процес програмування не представляє значних труднощів, так як реалізація ГА є циклічною процедурою застосування генетичних операторів до вихідної популяції хромосом.

Розглянемо деякі доступні пакети програмного забезпечення у цій галузі. В мережі Інтернет є у вільному доступі деякі матеріали для реалізації ГА. Наприклад пакет *GeneHunter*, розробленої американською фірмою Ward Systems Group у 1995 р. Даний пакет русифікований російською компанією Нейропроект, що є офіційним представником Ward Systems Group на території Росії. До складу пакета *GeneHunter* входять доповнення Microsoft Excel, що дозволяє користувачеві вирішувати оптимізаційні задачі з робочих аркушів Excel, Динамічна Бібліотека (Dynamic Link Library) функцій ГА, які можна викликати з таких мов програмування, як MS Visual Basic або C++, і демонстраційні приклади.

GeneHunter створює популяцію можливих рішень задачі. Індивідууми у цій популяції будуть описуватися хромосомами, якими є значення змінних у розв'язуваній задачі. *GeneHunter* переймається тим, дозволяючи вимирати індивідуумам, найменш задовольняє умовам завдання, і відбираючи для схрещування найбільш підходящих індивідуумів (тих, які дають найкраще рішення задачі) аналогічно відбору найбільш пристосованих індивідуумів у природі (селекція).

GeneHunter бере двох відповідних індивідуумів і виконує між ними частковий обмін генетичною інформацією (схрещування). Потомство схрещеної пари буде нести в собі частину ознак

від матері, а частину – від батька. Потомство часто має деякі невеликі відхилення від норми (мутація). Зазвичай такі мутації калічать і призводять до загибелі потомства, проте іноді вони покращують пристосовність індивідуума (як це було у випадку з'єднання пальців перетинкою). Подібним чином GeneHunter іноді викликає мутації у своїх популяціях.

Після того, як GeneHunter схрещує відповідних індивідуумів і робить мутації, настає зміна поколінь. Тепер популяція буде складатися з потомства плюс кілька старих індивідуумів, яким GeneHunter дозволяє вижити, оскільки вони є найкращими в популяції, і ми хочемо, щоб вони продовжували брати участь у схрещуванні. Такі найбільш підходящі індивідууми називаються елітою.

Через десятки і навіть сотні "поколінь" зрештою виникає популяція, в якій індивідууми вирішують задачу дуже добре. Фактично, найкращий (елітний) індивідуум дасть оптимальне або близьке до оптимального рішення.

В основному діалоговому вікні GeneHunter є комірка з цільовою функцією, обрані значення (максимум, мінімум або фіксована величина), до яких прагне алгоритм при пошуку, що підбираються параметри (хромосоми) разом з їх типом, список обмежень.

Наступне діалогове вікно дозволяє встановити параметри алгоритму. У вікні, крім відомих уже параметрів ймовірностей схрещування і мутації, потрібно вказати ступінь оновлення. Цей параметр визначає ту частину індивідуумів, яка не переходить у наступне покоління. Діапазон зміни цього параметра становить величину від 0 до 1. Наприклад, при значенні ступеня оновлення, що дорівнює 0,98, тільки 2% рядків перейдуть у наступне покоління, не наражаючись на схрещування і мутації, а 98% – загинуть. При розмірі популяції, що становить 50 рядків, тільки $50 \cdot 0,02 = 1$ індивідуум перейде, не змінюючись, у наступне покоління. У разі включення стратегії елітизму (ще одна опція у вікні

установки параметрів алгоритму) збережені рядки є найбільш підходящими (еліта).

У деяких випадках корисно спочатку запустити GeneHunter з відключеним елітизмом, щоб дозволити популяції розвиватися, не надаючи на процес сильного селекційного тиску. Через деякий час можна включити елітизм, щоб сконцентрувати процес пошуку оптимізації навколо кращого рішення.

Включення оператора різноманітності (установки мітки у лівому нижньому кутку) дозволяє виконати слабку форму мутації, що виробляє індивідууми зі злегка зміненими властивостями, на протипагу сильним змінам, які виконує стандартний оператор мутації. У деяких завданнях застосування цього оператора дає поліпшення процесу еволюції, в інших - ні, проте в більшості випадків його варто використовувати.

Резюмуючи результати цього розділу, можна зробити висновок про придатність генетичних алгоритмів, інспірованих біологічною еволюцією, до вирішення багатьох завдань, у тому числі і менеджменту, в яких необхідно отримати оптимальне рішення. Здійснюючи пошук по простору параметрів за допомогою набору можливих рішень, реалізованих хромосомами, можна, скоріше, досягти оптимального (або близького до нього) рішення, ніж у випадку застосування інших методів.

РОЗДІЛ 5

МОДЕЛЮВАННЯ СИСТЕМ АВТОМАТИЧНОГО РЕГУЛЮВАННЯ

5.1. Загальні відомості про моделювання САР

Системи автоматичного регулювання (САР) є частиною технологічного процесу. Внаслідок дії збурень, яким піддаються змінні процесу, порушується встановлений режим роботи об'єкта, що призводить до зміни вихідних змінних. Завдання регулювання об'єкта (об'єкт регулювання – це окремий технологічний апарат (або декілька одиничних технологічних апаратів), в якому протікає деякий процес) – усунення впливу збурень, тобто відновлення початкових значень вихідних змінних. Автоматичне регулювання – процес підтримки (стабілізації) або зміни за заданим законом вихідних змінних об'єкта за допомогою спеціальних пристроїв – автоматичних регуляторів.

Система автоматичного регулювання включає взаємодіючі між собою об'єкт регулювання і засоби автоматичного регулювання і являє собою динамічну систему, описувану диференціальними рівняннями.

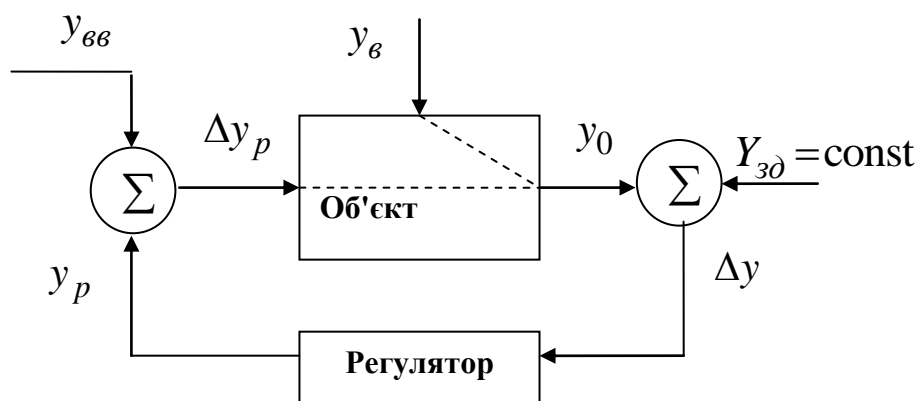


Рис.5.1. Принципова схема замкнутого контуру САР за відхиленням – система стабілізації

Система регулювання містить дві взаємодіючі ланки: об'єкт регулювання і регулятор, причому найчастіше система «об'єкт – регулятор» становить замкнутий контур регулювання (рис.5.1) за принципом зворотного зв'язку (одноконтурні САР).

Одним з основних завдань регулювання є стабілізація технологічних (режимних) параметрів процесу на заданих рівнях. Потрібне значення параметра $y_{zd} = \text{const}$ подається на підсумковий пристрій регулятора, де порівнюється з дійсним значенням параметра $y_0 = y_0(t)$ на виході об'єкта. Значення називається регульованою змінною. Різниця $\Delta y = (y_0 - y_{zd})$ – сигнал неузгодження виникає як наслідок деяких збурюючих впливів $y_{\text{вв}}$.

У відповідь на сигнал неузгодженості регулятор виконує регулюючий вплив y_p , що разом зі збуренням $y_{\text{вв}}$ надходить на пристрій, що сумує, утворюючи різницю $\Delta y_p = (y_{\text{вв}} - y_p)$. Величину і знак часу компенсувати повністю (або до заданого рівня) дії збурень, тобто забезпечити $\Delta y_p = 0$ (або $\Delta y \leq \varepsilon$). Описаний принцип називається регулюванням за неузгодженістю або за відхиленням, а використана САР називається стабілізуючою. Як збурення виступають такі фактори: 1) зовнішні технологічні параметри $y_{\text{вв}}$ – порушення режиму роботи об'єкта (наприклад, зміна температури, об'ємної швидкості подачі тощо); 2) характеристики навколишнього середовища – зміна зовнішніх умов (наприклад, коливання температури, вологості тощо); 3) внутрішні параметри об'єкта, наприклад непередбачені витоки сировини або продукту, зміни умов теплопередачі через стінку апарату тощо.

Системами автоматичної стабілізації не вичерпуються можливості одноконтурних САР технологічних процесів. Наприклад, у системах програмного регулювання регульована величина змінюється відповідно до заданої функції $y_{zd} = F(t)$, а

слідкуючі системи регулювання (рис. 5.2) змінюють регульований параметр y_0 залежно від значень якої-небудь змінної технологічного процесу $y_{зд} = F(x_T)$.

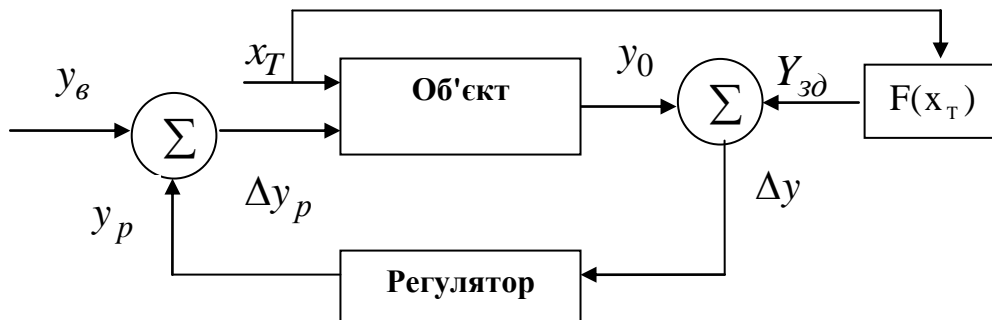


Рис.5.2. Принципова схема одноконтурної САР за відхиленням – слідкуюча система

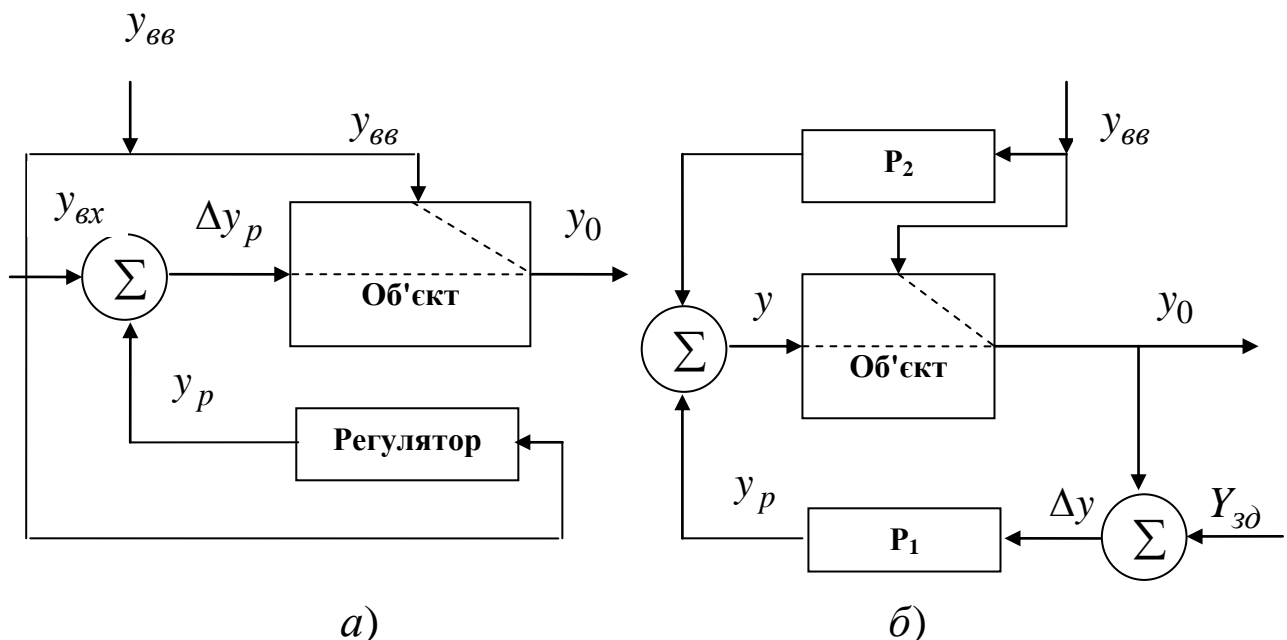


Рис.5.3. Принципова схема одноконтурної САР за збуренням а) і принципова схема комбінованої САР б)

У САР використовується також принцип регулювання за збуренням. У результаті застосування цього принципу отримують системи регулювання розімкнутого типу без зворотного зв'язку

(рис.5.3, а). Збурення $y_{вв}$ одночасно сприймається об'єктом і регулятором. Регулятор повністю усуває вплив збурення шляхом видачі відповідного регулюючого впливу y_p на вхідний параметр $y_{вх}$.

Об'єднання принципів регулювання за неузгодженістю і за збуренням, реалізовано в комбінованих САР, знайшло широке застосування (рис.5.3, б). Спрощена модель САР (рис.5.4) містить математичний опис об'єкта і регулятора.



Рис.5.4. Схема спрощеної моделі одноконтурної САР

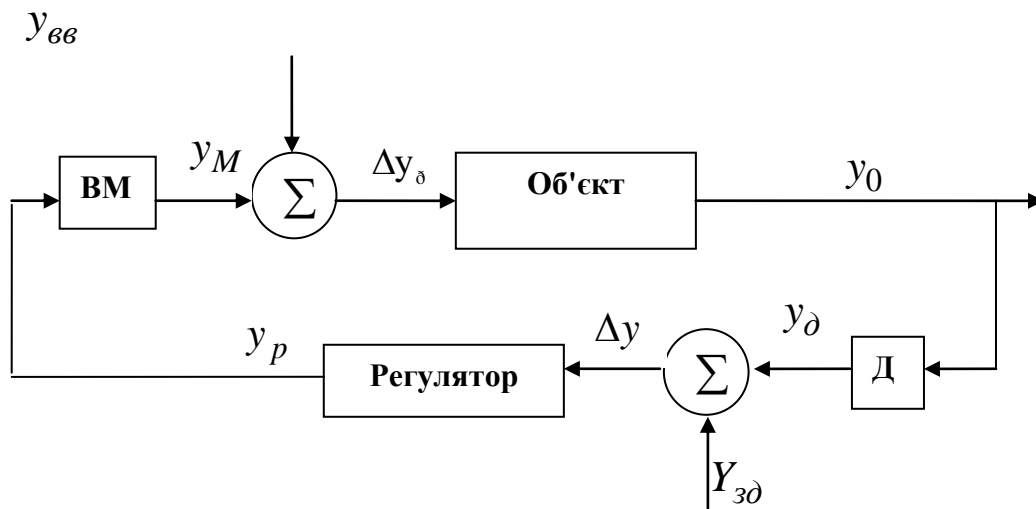


Рис.5.5. Схема одноконтурної САР, що враховує допоміжні засоби регулювання

Для більш точних розрахунків і досліджень САР за допомогою математичного моделювання необхідно враховувати вплив на поведінку системи допоміжних засобів регулювання (рис. 5.5), до яких відносяться датчики D , що вимірюють значення регульованого сигналу (витрата, рівень, концентрація, температура тощо); виконавчі механізми ВМ (наприклад, електродвигуни), що приводять до дії регулюючий орган (вентиль, клапан, дросельна заслінка тощо). Математичний опис змінних ланок системи входить у загальну модель САР.

Таким чином, поведінка САР у кінцевому рахунку визначається динамічними характеристиками складових її елементів (ланок). При моделюванні САР на ЕОМ за допомогою оболонки MatLab (пакет розширення Simulink) прийняті способи опису ланок системи з допомогою диференціальних рівнянь і з допомогою передавальних функцій.

5.2. Моделювання типових ланок САР

До складу системи автоматичного регулювання входять різні елементи, які розрізняють за принципом дії, конструктивним рішенням тощо. Однак, незважаючи на велику різноманітність, ці елементи вдається класифікувати за їх динамічними властивостями. Реальні елементи системи автоматичного регулювання можна з достатньою точністю апроксимувати окремими типовими елементарними ланками або комбінаціями декількох таких ланок.

Типові ланки САР мають ряд важливих особливостей і властивостей:

- 1) параметри ланки зосереджені і не змінюються за часом;
- 2) ланка має властивість направленої дії, тобто передає сигнал тільки в одному напрямку – з входу на вихід;
- 3) динамічні властивості ланки характеризуються кривою розгону, яка представляє собою залежність вихідної змінної

ланки $x_{вих}$ від часу в разі подачі на вхід ланки одиничного ступінчастого впливу $x_{вх}$ при заданих початкових умовах;

4) ідеалізована характеристика ланки описується лінійним диференціальним рівнянням n -го порядку або відповідною передавальною функцією.

Розглянемо типові ланки САР та їх моделі.

1. **Безінерційна або підсилювальна ланка** (передача вхідного сигналу на вихід відбувається практично миттєво).

Рівняння ланки записується у вигляді:

$$x_{вих}(t) = k_p x_{вх}(t),$$

де k_p – коефіцієнт передачі ланки.

Для моделювання безінерційної ланки в пакет розширення Simulink використовують підсилювач Gain (рис.5.6).

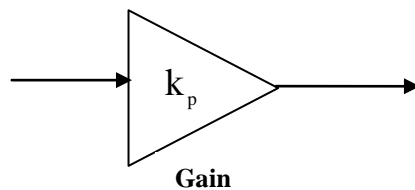


Рис. 5.6. Схема безінерційної ланки

2. **Інерційна, або аперіодична, ланка 1-го порядку** має ємність по речовині (або енергії) й опір потоку цієї речовини (або енергії). Вона описується лінійним диференціальним рівнянням 1-го порядку з постійними коефіцієнтами:

$$T \frac{dx_{вих}}{dt} + x_{вих} = kx_{вх},$$

де T – постійна часу аперіодичної ланки, рівна добутку ємності на опір; k – коефіцієнт передачі ланки.

Інерційна ланка 1-го порядку моделюється на інтеграторі (рис.5.7), охопленому зворотним зв'язком. Для моделювання

рівняння інерційної ланки перепишемо відносно похідної попередньо поділивши на T :

$$\frac{dT_{вих}}{dt} = -\frac{1}{T}x_{вих} + \frac{k}{T}x_{вх}.$$

В якості вхідного сигналу використовують ступінчасте збурення. Реакцію системи на ступінчасте збурення можна подивитися за допомогою осцилографа Score.

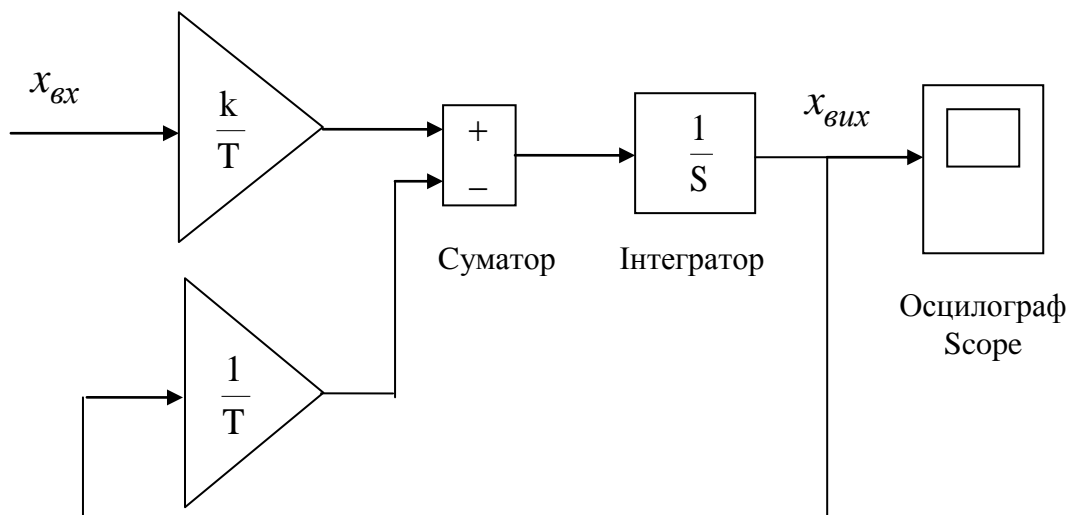


Рис. 5.7. Схема моделі інерційної ланки 1-го порядку

3. Інерційна чи аперіодична ланка 2-го порядку виходить при послідовному з'єднанні двох фізичних елементів, кожен з яких має ємність і обмінюється речовиною (або енергією) через деякий опір.

Динаміка цієї ланки описується диференціальним рівнянням 2-го порядку:

$$T_2^2 \frac{d^2 x_{вих}}{dt^2} + T_1 \frac{dx_{вих}}{dt} + x_{вих} = kx_{вх}$$

за умови що $T_1 \gg 2T_2$.

Для моделювання аперіодичної ланки 2-го порядку рівняння приводять до вигляду:

$$\frac{d^2 x_{вих}}{dt^2} = \frac{k}{T_2^2} x_{вх} - \frac{1}{T_2^2} x_{вих} - \frac{T_1}{T_2^2} \frac{dx_{вих}}{dt},$$

тобто схема моделі буде складатися з суматора з трьома входами і трьома блоками Gain, двох послідовно з'єднаних інтеграторів та осцилографа Scope.

Коливальна ланка описується тим же рівнянням 2-го порядку і відрізняється від інерційної співвідношенням між постійними часу: $T_1 < 2T_2$.

5.3. Моделювання законів регулювання автоматичних регуляторів безперервної дії

Регулятор безперервної дії – це пристрій, який при безперервній подачі регульованої величини на його вхід безперервно формує на своєму виході регулюючий сигнал, відповідний заданим законом регулювання (рис.5.8). При моделюванні закон регулювання представляється деяким рівнянням регулятора. Розглянемо основні типи регуляторів безперервної дії та їх моделі.

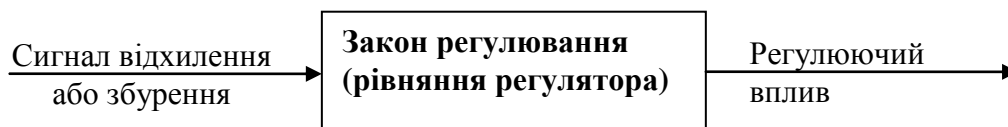


Рис.5.8. Схема регулятора безперервної дії

1. **Пропорційний або статичний регулятор** (П-регулятор). При відхиленні регульованої величини від її заданого значення цей регулятор видає команду на переміщення регулюючого органу, пропорційне відхиленню; таким чином, величина сигналу на виході регулятора пропорційна вхідному сигналу. Закон регулювання записується у вигляді:

$$y(t) = k_p x(t),$$

де $y(t)$ – регулюючий вплив регулятора; k_p – коефіцієнт передачі регулятора; $x(t)$ – сигнал неузгодження.

Коефіцієнт k_p називається коефіцієнтом підсилення регулятора. Він обчислюється як відношення зміни сигналу на виході регулятора, прийняте за 100 %, до сигналу неузгодженості, що викликав цю зміну.

Для моделювання П-регулятора у пакет розширення Simulink використовують підсилувач Gain.

2. **Інтегральний або І-регулятор** працює згідно з рівнянням:

$$y(t) = k_i \int x(t) dt,$$

де k_i – коефіцієнт передачі регулятора, що характеризує швидкість роботи виконавчого механізму.

Закон регулювання показує, що швидкість регулюючого впливу dy/dt прямо пропорційна сигналу неузгодженості $k_i x$. Модель І-регулятора в Simulink складається з послідовно з'єднаних блоків Gain з коефіцієнтом k_i та інтегруючого блока.

3. **Пропорційно-інтегральний регулятор** (ПІ-регулятор) утворюється паралельним з'єднанням П-регулятора та І-регулятора. Закон регулювання виражається рівнянням:

$$y(t) = k_p x + k_i \int x dt = k_p \left(x + \frac{1}{T_i} \int x dt \right),$$

де k_p – статичний коефіцієнт передачі; T_i – час ізодрома.

Константа $1/T_i$, що входить до інтегральної складової, має розмірність 1/сек. Її величина характеризує час, протягом якого відбувається автоматична перестановка регулюючого органу з одного крайнього положення в інше за рахунок інтегральної складової регулюючого впливу. Таким чином, регулюючий вплив

ПІ-регулятора пропорційний сумі відхилення регульованої величини в часі від заданої та інтеграла цього відхилення.

Модель ПІ-регулятора (рис.5.9) складається з 4-х блоків: двох підсилювачів, інтегратора і суматора, на виході якого формується регулюючий вплив ПІ-регулятора.

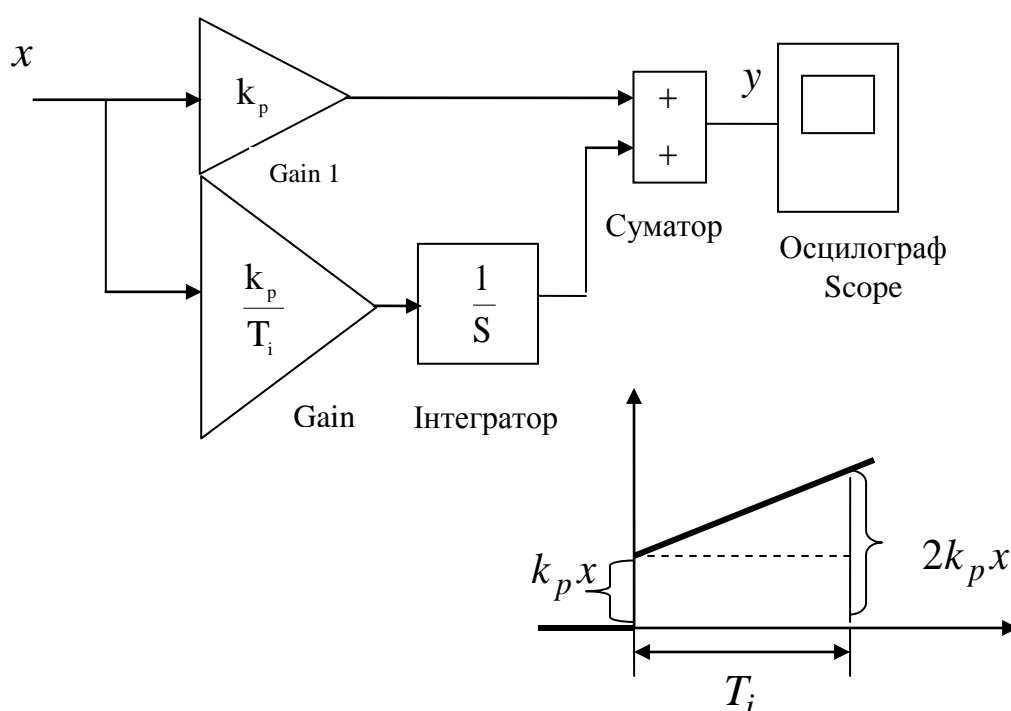


Рис.5.9. Схема моделі ПІ-регулятора

4. **Диференціальний регулятор** (Д-регулятор). Величина вихідного (регулюючого) сигналу прямо пропорційна швидкості зміни вхідного сигналу:

$$y(t) = k_d \frac{dx}{dt}.$$

Д-регулятор зазвичай не використовується самостійно, а входить як складова частина до пропорційно-диференціального (ПД-регулятор) і пропорційно-інтегрально-диференційного регулятора (ПІД-регулятор).

5. **Пропорційно-інтегральний-диференціальний** регулятор (ПІД-регулятор) називають ще ізодромним регулятором з початком. Закон його регулювання описується рівнянням такого вигляду:

$$y(t) = k_p \left(x + \frac{1}{T_i} \int x dt + T_d \frac{dx}{dt} \right),$$

де k_p – коефіцієнт передачі регулятора; T_i – час ізодрома; T_d – час перетворення.

Модель ПІД-регулятора (рис. 5.10) складається з п'яти блоків: трьох підсилювачів, інтегратора, диференціатора і суматора, на виході якого формується регулюючий вплив ПІД-регулятора.

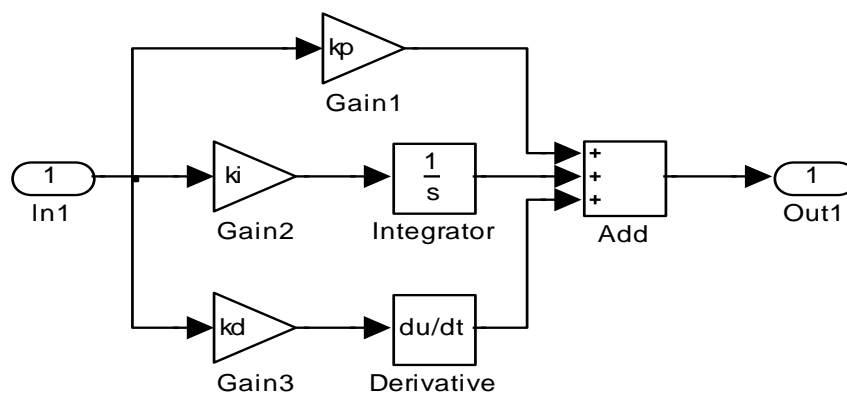


Рис.5.10. Схема моделі ПІД-регулятора

5.4. Передавальні функції

Поняття передавальної функції склалося в теорії систем автоматичного регулювання для опису динамічних властивостей лінійних ланок, які мають зосереджені, кінцеві і постійні в часі параметри. За визначенням передавальною функцією лінійної ланки $W(p)$ називається відношення миттєвих сигналів на виході $y(p)$ і на вході $x(p)$ ланки в операторній формі:

$$w(p) = \frac{y(p)}{x(p)}$$

Диференціальні рівняння завжди можна перетворити на відповідну передавальну функцію. Наведені вище диференціальні рівняння можуть бути записані в операторній формі шляхом заміни похідної d/dt символом p . Це дозволяє звести рішення диференціального рівняння до розв'язування алгебраїчного, що набагато простіше. Таке перетворення широко використовується в теорії автоматичного регулювання.

Представимо рівняння інерційної ланки

$$T \frac{dx_{вих}}{dt} + x_{вих} = kx_{вх}$$

в операторній формі

$$Tp x_{вих}(p) + x_{вих}(p) = kx_{вх}(p)$$

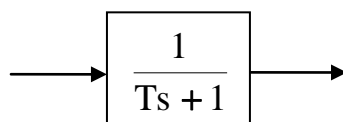
або

$$x_{вих}(p)(Tp + 1) = kx_{вх}(p).$$

Запишемо передавальну функцію:

$$W(p) = \frac{x_{вих}(p)}{x_{вх}(p)} = \frac{k}{Tp + 1},$$

де k – коефіцієнт передачі ланки; T – постійна часу ланки. В Simulink ця передавальна функція має вигляд:



Для рівняння аперіодичної ланки 2-го порядку

$$T_2^2 \frac{d^2 x_{вих}}{dt^2} + T_1 \frac{dx_{вих}}{dt} + x_{вих} = kx_{вх}$$

операторна форма запису має вигляд:

$$T_2^2 p^2 x_{\text{вих}}(p) + T_1 p x_{\text{вих}}(p) + x_{\text{вих}}(p) = k x_{\text{вх}}(p).$$

Передавальна функція:

$$W(p) = \frac{x_{\text{вих}}(p)}{x_{\text{вх}}(p)} = \frac{k}{T_2^2 p^2 + T_1 p + 1}.$$

Передавальні функції І-регулятора, ІІ-регулятора і ІІД-регулятора відповідно, мають вигляд:

$$W(p) = \frac{k_i}{p}; \quad W(p) = \frac{k_i + k_p p}{p}; \quad W(p) = \frac{k_i + k_p p + k_d p^2}{p}.$$

Таким чином будь-ланку можна представити у вигляді передавальної функції. У пакеті Simulink моделюється блоком передавальної характеристики **Transfer Fcn** (задає передавальну функцію у вигляді відношення поліномів, порядок чисельника не повинен перевищувати порядок знаменника). На рис. 5.11 показаний приклад моделювання коливальної ланки за допомогою блоку Transfer Fcn.

При складанні і перетворення структурної схеми САР передавальні функції більш зручні, ніж диференціальні рівняння. Ланки системи регулювання можуть бути з'єднані один з одним послідовно, паралельно, ланка може бути охоплена зворотним зв'язком.

Передавальна функція послідовно з'єднаних ланок дорівнює добутку передаточних функцій усіх ланок:

$$w(p) = w_1(p)w_2(p)\dots w_n(p)$$

Передавальна функція паралельно з'єднаних ланок дорівнює сумі передавальних функцій ланок:

$$w(p) = w_1(p) + w_2(p) + \dots + w_n(p)$$

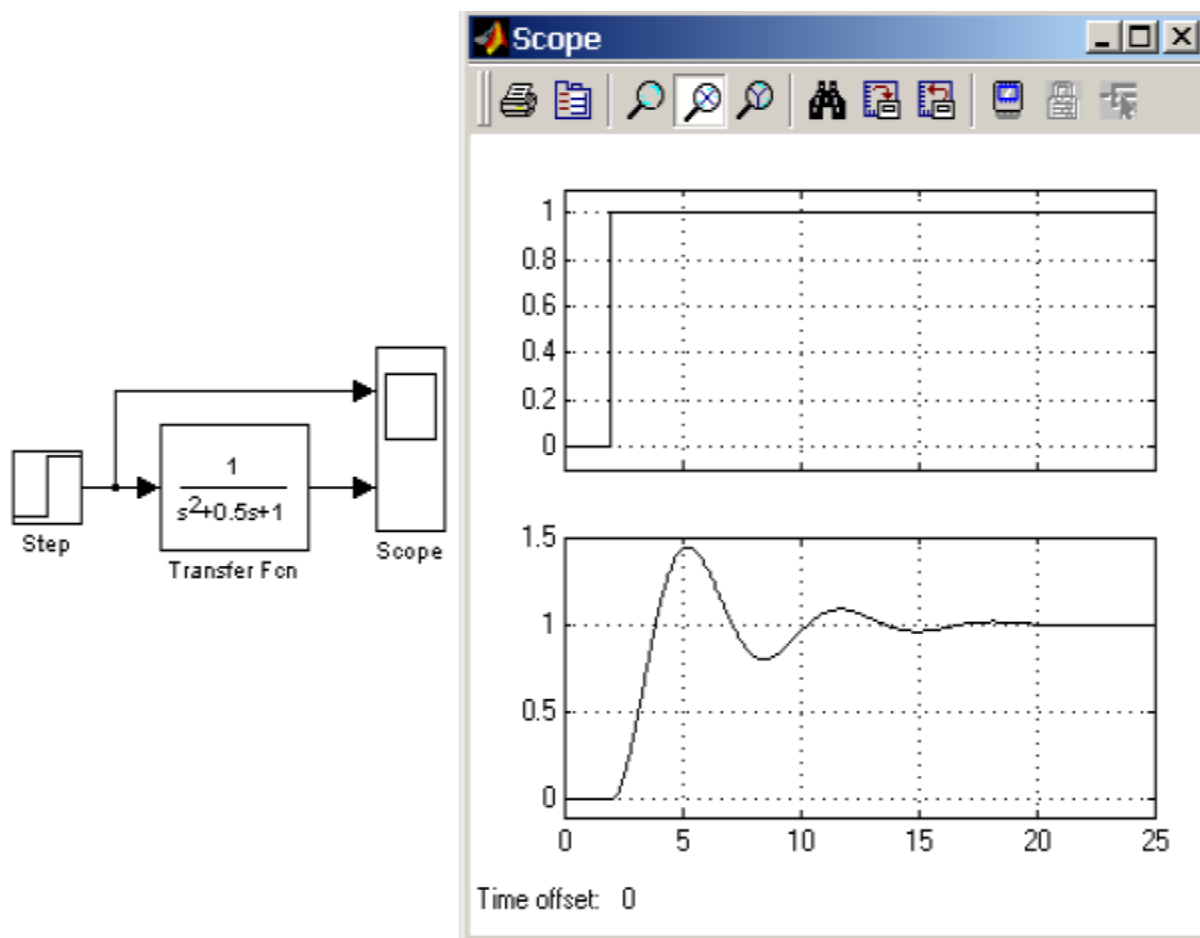


Рис. 5.11. Приклад моделювання коливальної ланки

Передавальна функція замкнутої системи (ланка зі зворотним зв'язком) визначається виразом:

$$w(p) = \frac{w(p)}{1 + w(p)}$$

Приклад 5.1. Регулювання концентрації суміші двох рідин у проточному апараті повного перемішування. У ємності проточного типу (рис.5.12) готують розчин заданої концентрації $C_{a,зд}$ шляхом перемішування двох потоків рідини.

Об'ємні витрати і концентрації вихідних компонентів відповідно позначені $v_1, C_{a,1}, v_2, C_{a,2}, v_1=1; v_2=1; V=10$. Відхилення концентрації готової суміші C_a від заданого значення

виникає через збурення по навантаженню і регулюється зміною потоку v_2 . Концентрації $C_{a,1}$ і $C_{a,2}$ постійні $C_{a,1}=1$; $C_{a,2}=1$. Таким чином, розглядається одноконтурна система регулювання по відхиленню (рис. 5.12). Отримати диференціальне рівняння (або передавальну функцію об'єкта регулювання. Скласти структурну схему моделі системи регулювання для П-, І-, ІІ- і ІІД-регуляторів. Отримати криві перехідного процесу. Моделювання провести в пакеті Simulink.

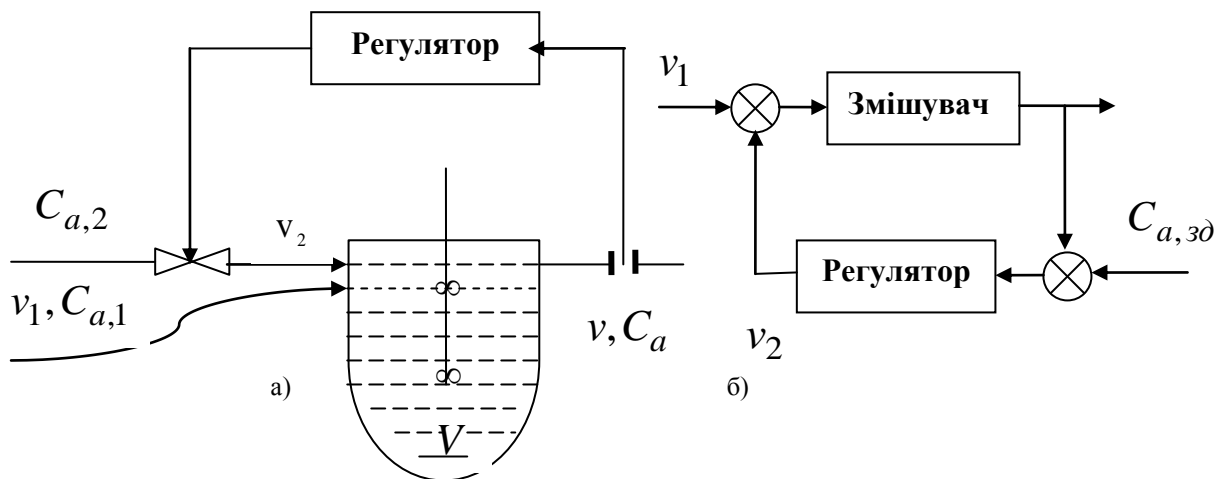


Рис.5.12. Регулювання концентрації у змішувачі проточного типу:
а) схема регулювання; б) структурна схема САР

Складемо математичний опис ланок системи регулювання. Диференціальне рівняння матеріального балансу змішувача має вигляд:

$$V \frac{dC_a}{dt} = v_1 C_{a,1} + v_2 C_{a,2} - v C_a,$$

де $v = v_1 + v_2$.

Поділимо всі члени рівняння на v і запишемо його у вигляді:

$$T \frac{dC_a(t)}{dt} + C_a(t) = k_1 v_1 + k_2 v_2,$$

де $k_1 = \frac{C_{a,1}}{v}$; $k_2 = \frac{C_{a,2}}{v}$ – коефіцієнти підсилення змішувача;

$T = \frac{V}{v}$ – постійна часу змішувача.

Таким чином, змішувач являє собою аперіодичну одноємнісну ланку. Так як відбір з ємності повинен бути постійним v , то будь-яка зміна об'ємної витрати v_1 викликає відповідно рівний приріст потоку v_2 , тобто $v_1 = -v_2$. Проведемо моделювання за допомогою пакета Simulink із застосуванням ПД-регулятора (датчик і виконавчий механізм включаємо в рівняння ідеального регулятора безперервної дії). Таким чином рівняння

$$\frac{dC_a(t)}{dt} = \frac{k_1 v_1}{T} + \frac{k_2 v_2}{T} - \frac{1}{T} C_a$$

та ПД-регулятор описує систему регулювання (рис.5.13).

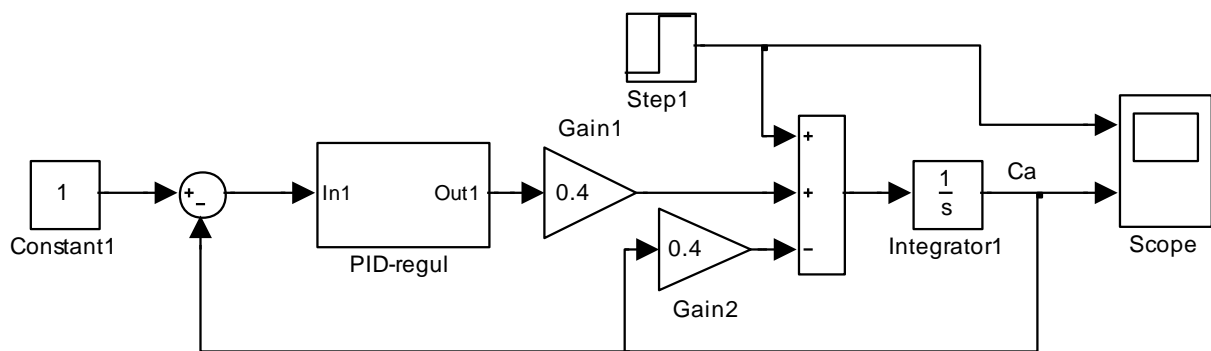


Рис. 5.13. Модель САР змішувача

Функція збурюючої дії представляється у вигляді ступінчастої зміни об'ємної витрати v_1 . Вихідні дані: V – об'єм змішувача; об'ємні витрати $v_{1,0}$ і $v_{2,0}$; характер і параметри

збурюючої дії $v_1(t)$; настроювання регуляторів k_p, k_i, k_d ; задає вплив на регулятор $C_{a,зд}$; початкова концентрація в змішувачі $C_a(0)$ (до нанесення збурення).

PID-regul створений як підсистема з допомогою блоку SubSystem (блок-схема в Simulink може мати структуру «матрьошки»). Відкривши вікно розділів бібліотеки Simulink, необхідно перетягнути мишею в це вікно блок SubSystem з розділу Ports&Subsystems. Потім наводимо курсор миші на цей блок і двічі клацнемо мишею. Відкриється вікно підсистеми. Це вікно майже порожнє – у ньому є лише блоки In1 і Out1, сполучені стрілкою.

Перемістивши це вікно у зручне положення, виділимо мишею порт In1 і з'єднання, знищимо їх командою Clear у позиції меню Edit вікна підсистеми. Відкривши поруч вікно розділів бібліотеки Simulink, можна почати будувати підсистему (рис.5.14). Якщо підсистема містить помилки, то їх можна згодом виправити.

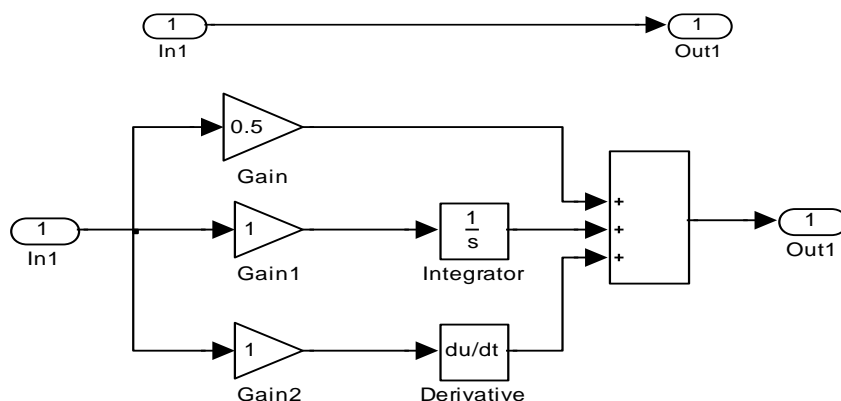


Рис.5.14. Побудована підсистема

На рис.5.15 представлені графіки ступеневого збурення (верхній) і перехідного процесу регулювання (нижній), які побудовані за допомогою осцилографа Scope.

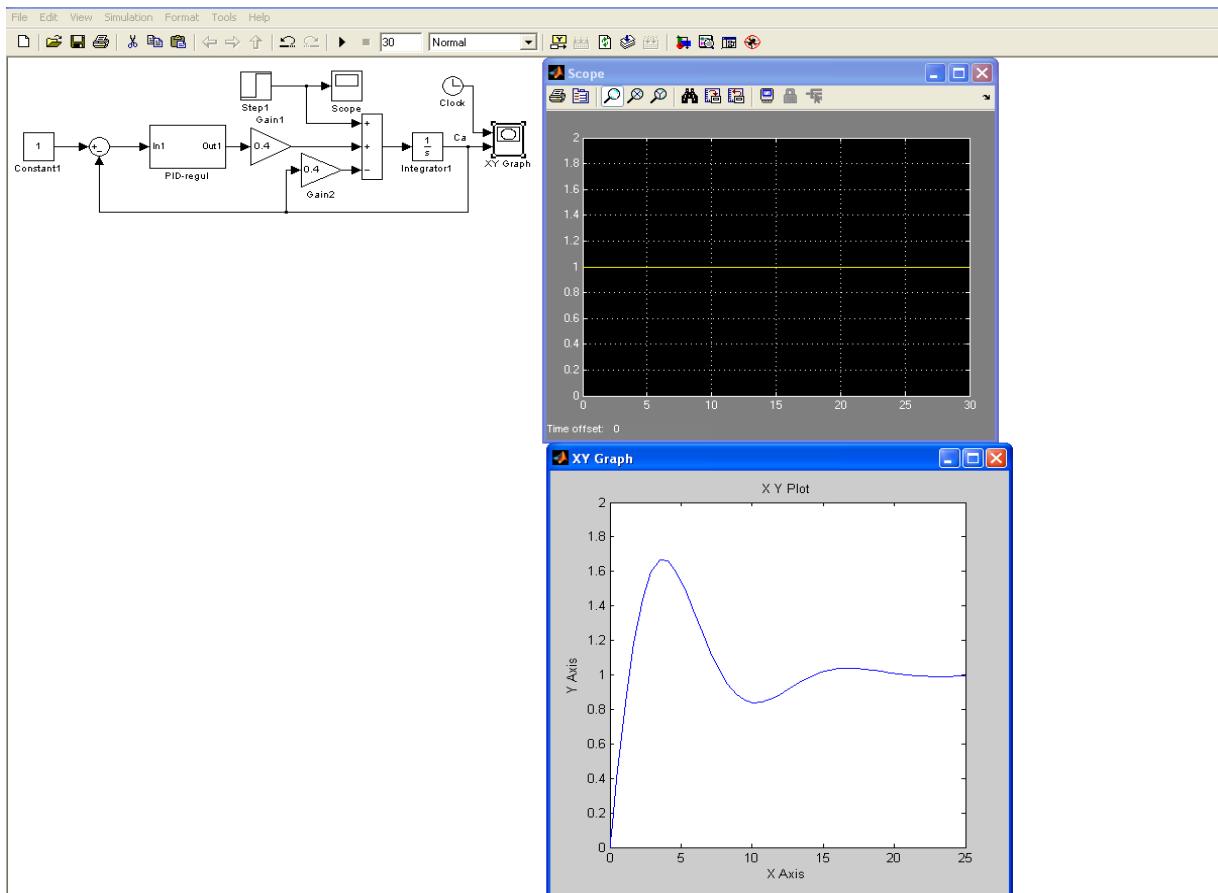
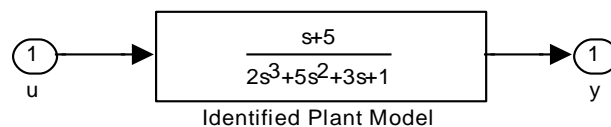


Рис.5.15. Графік перехідного процесу регулювання при ступінчастому збуренні

Нарис. 5.16 представлена система регулювання з ПІД-регулятором, де об'єктом регулювання виступає ланка з передавальною функцією:



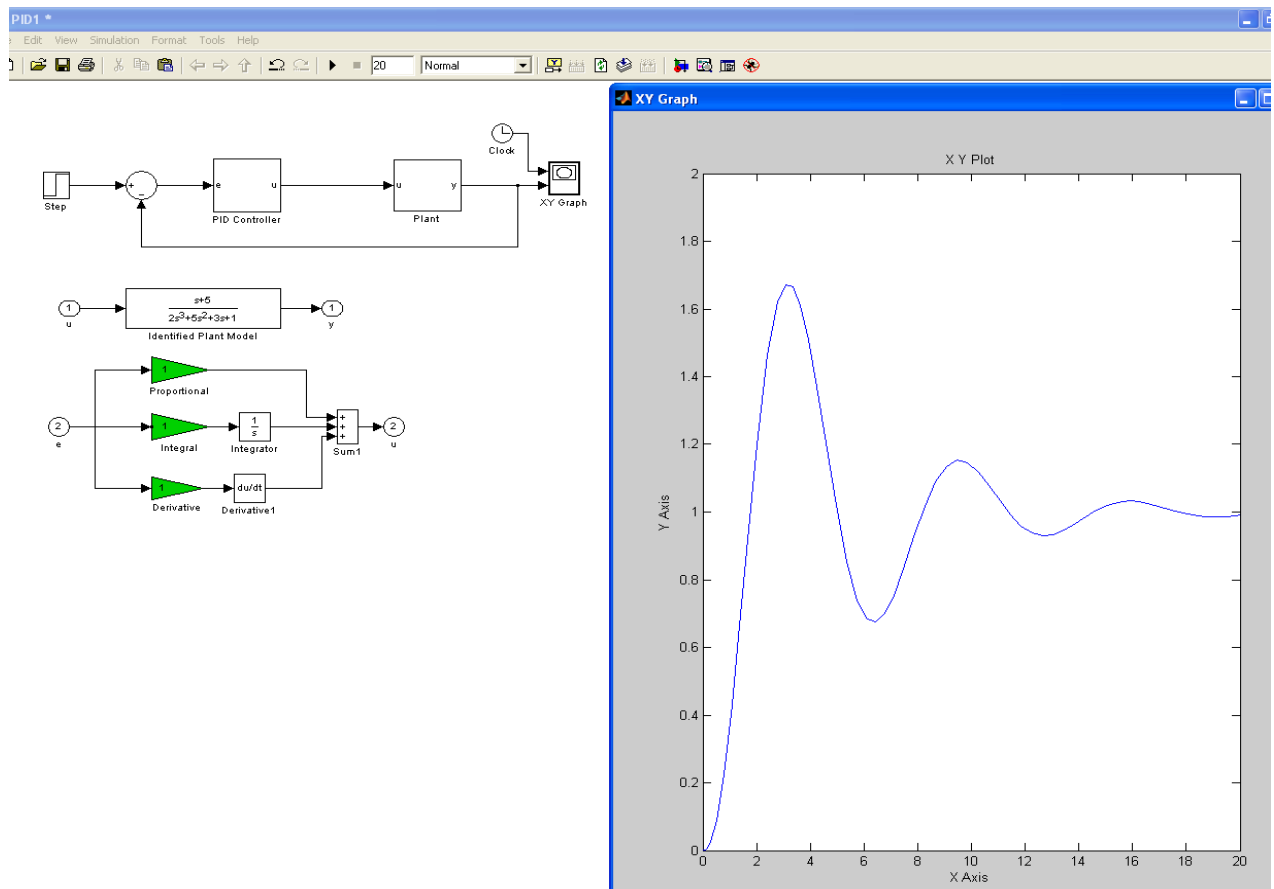


Рис. 5.16. Модель об'єкта оптимізації з ПІД-контролером і графік перехідного процесу

Вправи для самостійного розв'язання

Вправа 1. В одноконтурну систему регулювання змішувача (див. приклад вище) включений П-регулятор. Допустима величина залишкового відхилення $\delta = 0,05C_{a,уст}$. Знайти коефіцієнт налаштування k_p , що задовольняє необхідній якості для ступінчастого зовнішнього впливу. Побудувати графіки залежності залишкового відхилення залежно від коефіцієнта k_p .

Вправа 2. У контур регулювання змішувача включений І-регулятор.

А). Потрібно вибрати налаштування регулятора k_i таким чином, щоб динамічне відхилення не перевищувало заданого значення:

$C_{a,дин} \leq 0,4C_{a,уст}$. Побудувати залежність максимального динамічного відхилення від значень налаштування k_i для ступінчастого збурення.

Б). Вибір настройки І-регулятора визначається часом регулювання, який задається.

Вправа 3. Регулювання змішувача здійснюється ПІ-регулятором. Потрібно знайти такі параметри регулятора k_p, k_i , щоб інтегральна оцінка якості була найменша, тобто $\min \Delta = \int C_a^2(t) dt$.

Схема оцінки якості підключається до виходу інтегруючого блоку.

Вправа 4. Скласти модель і досліджувати перехідний процес з інтегральною оцінкою якості для одноконтурної системи регулювання (див. вправу 3) з урахуванням динамічних властивостей чутливого елемента (сенсора) і виконавчого механізму. Об'єктом регулювання є змішувач, описаний вище. Передавальні функції датчика концентрації та виконавчого механізму мають вигляд:

$$W_{\partial}(p) = \frac{k_{\partial}}{T_{\partial}p + 1}; W_{вм}(p) = \frac{k_{м}}{T_{м}p + 1}$$

Структурна схема САР змішувача з датчиком і виконавчим механізмом наведена на рис. 5.17.

Вправа 5. Для отриманої S-моделі у вправі 5 провести дослідження одноконтурної системи регулювання у двох напрямках:

- 1) як впливають на якість регулювання динамічні властивості датчика та виконавчого механізму;
- 2) для датчика та виконавчого механізму встановити найкращий закон регулювання, тобто вибрати тип і параметри настройки регулятора за інтегральним критерієм якості.

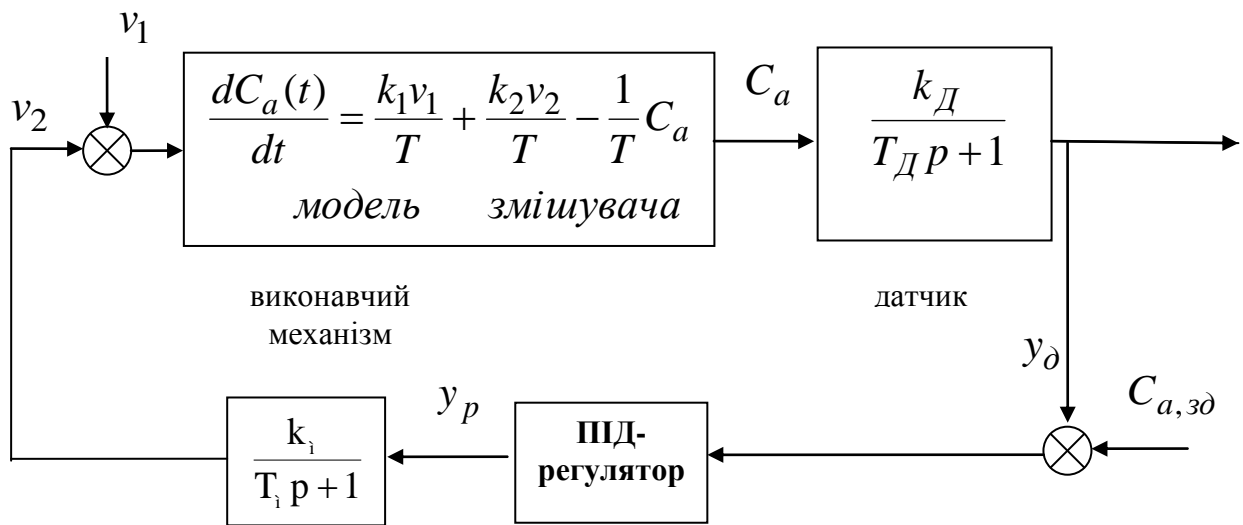


Рис. 5.17. Структурна схема САР змішувача з датчиком і виконавчим механізмом

Вправа 6. В одноконтурну систему автоматичного регулювання змішувача включити нелінійність типу «зона нечутливості» (блок dead zone) регулятора, як показано на рис.5.18. Величину зони нечутливості вимірювати в межах 5–10% від $u(\infty)$ кривої розгону об'єкта. Зняти перехідні процеси з об'єкта регулювання для двох-трьох значень зони нечутливості при незмінній настройці регулятора і постійної величини зовнішнього впливу. Зіставити графіки перехідних процесів, а також отримати інтегральні оцінки якості регулювання.

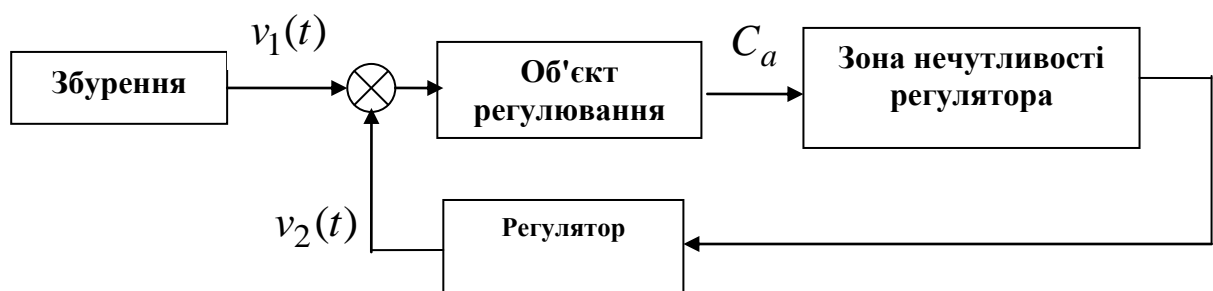


Рис. 5.18. Схема включення до одноконтурної САР нелінійності типу «зона нечутливості»

Вправа 7. В одноконтурну систему автоматичного регулювання змішувача включити нелінійність типу «обмеження по модулю» (блок satu-ration), а потім «гістерезис» (блок relay) виконавчого механізму, як показано на рис. 5.19. Зняти криві регулювання при незмінній настройці регулятора, але при різній величині ступінчастого впливу. Зіставити інтегральні оцінки якості регулювання.



Рис.5.19. Схема включення до одноконтурної САР нелінійності «обмеження по модулю» або «гістерезис»

ЛІТЕРАТУРА

1. *Прангишвили И. В.* Системный подход и общесистемные закономерности / И. В. Прангишвили. — Москва: "СИНТЕГ", 2000. — 520 с.

2. *Дорохов И. Н.* Системный анализ процессов химической технологии. Интеллектуальные системы и инженерное творчество в задачах интенсификации химико-технологических процессов и производств / И.Н. Дорохов, В. В. Меньшиков. — Москва: "Наука", 2005. — 582 с.

3. *Кафаров В. В.* Системный анализ процессов химической технологии. Основы стратегии / В. В. Кафаров, И. Н. Дорохов. — Москва: Наука, 1976. — 500 с.

4. *Кафаров В. В.* Системный анализ процессов химической технологии. Процессы полимеризации / В.В. Кафаров, И.Н. Дорохов, Л.В. Дранишников. — Москва: Наука. 1991. — 352 с.

5. *Химмельблау Д.* Анализ процессов статистическими методами / Д. Химмельблау.—М.: Издательство "Мир", 1973.— 950 с.

6. *Френкс Р.* Математическое моделирование в химической технологии. Пер. с англ. / Р. Френкс. — М.: Химия, 1971. — 272 с.

7. *Заде Л. А.* Понятие лингвистической переменной и его применение к принятию приближенных решений / Л.А. Заде. — М.: "Мир", 1976. — 165 с.

8. *Кричевский М. Л.* Интеллектуальные методы в менеджменте. Нейронные сети. Нечеткая логика, Генетические алгоритмы. Динамические системы / М. Л. Кричевский. — Питер: ЗАО Издательский дом "Питер", 2005. — 304 с.

9. *Чернавский Д. С.* Динамическая модель поведения общества. Синергетический подход к экономике. Новое в синергетике: Взгляд в третье тысячелетие / Д.С. Чернавский, Н. И. Старков, А. В. Щербаков. — М.: Наука, 2002. — С.239—291.

10. *Кириллюк И. Л.* Экономическая динамика Мир-Системы: взаимодействие стран с разным уровнем развития // История и математика: Модели и теории / И. Л. Кириллюк, С. Ю. Малков, А. С. Малков. — М.: Издательство ЛКИ, 2008. — С. 102—119.

11. *Осовский С.* Нейронные сети для обработки информации / С. Осовский. — Москва: Финансы и статистика, 2004. — 344с.

12. *Медведев В. С.* Нейронные сети. MatLab 6: Учебно-справочное издание. Пакеты прикладных программ/ В. С. Медведев, В. Г. Потемкин. — М.: ДИАЛОГ– МИФИ, 2002. — 496 с.

13. *Дьяконов В. П.* MATLAB 6.5 SP1/7/7 SP1/7 SP2 + Simulink 5/6 Инструменты искусственного интеллекта и биоинформатики / В. П. Дьяконов, В. В. Круглов. — М.: Солон–Пресс, 2006. — 454 с.

14. *Барский А. Б.* Нейронные сети: распознавание, управление, принятие решений/ А. Б. Барский. — М.: "Финансы и статистика", 2004. — 175 с.

15. *Круглов В. В.* Нечеткая логика и искусственные нейронные сети / В.В Круглов, М.И. Ли, Р. Ю. Голунов. — М.: Физматлит, 2001. — 310 с.

16. *Усков А. А.* Интеллектуальные технологии управления. Искусственные нейронные сети и нечеткая логика / А. А. Усков, А. В. Кузьмин. — Москва: "Горячая линия – Телеком", 2004. — 144 с.

17. *Дьяконов В. П.* Математические пакеты расширения MATLAB. Спец. справочник / В. П. Дьяконов, В. В. Круглов. — СПб.: Издательский дом "Питер", 2001. — 475 с.

18. *Липатова С. В.* Сборник задач по курсу "Интеллектуальные информационные системы": учебное пособие/ С. В. Липатова. — Ульяновск: Издательство Ульяновского государственного университета «УлГУ», 2010. — 64 с.

19. *Богатиков В. Н.* Построения систем управления на основе нейронных сетей: учебное пособие / В. Н. Богатиков,

Л. В. Дранишников, А. Е. Пророков. — Петрозаводск: Издательство Петрозаводского государственного университета, 2011. — 42 с.

20. *Тэрано Т.* Прикладные нечеткие системы / Т. Тэрано, К. Асаи, М. Сугено. — Москва: Мир, 1993. — 365 с.

21. *Леоненков А.* Нечеткое моделирование в среде MATLAB и fuzzyTECH / А. Леоненков. — Санкт-Петербург: БХВ — Петербург 2003. — 720 с.

22. *Богатиков В. Н.* Основы построения нечетких систем управления: учебное пособие / В. Н. Богатиков, Л.В. Дранишников, А. Е. Пророков. — Петрозаводск: Издательство Петрозаводского государственного университета, 2011. — 74 с.

23. Кузнецов П. Б. Основы нечеткой математики (теория нечетких множеств): учебное пособие / П. Б.Кузнецов, Ю. К. Оленникова. — Ярославль: Издательство Ярославского государственного университета, 2003. — 155 с.

24. *Васильев В. И.* Интеллектуальные системы управления. Теория и практика: учебное пособие / В. И. Васильев, Б. Г. Ильясов. — Москва: Издательство "Радиотехника", 2009. — 388 с.

25. *Де Янг К.* Эволюционные вычисления: новейшие достижения и нерешенные проблемы. Обзорение прикладной и промышленной математики / К. Де Янг. — М.: Научное издательство "ТВП", 1996. —Т.3. —Вып.5. — 179 с.

26. *Бураков М. В.* Генетический алгоритм: Теория и практика: учебное пособие / М. В. Бураков. — Санкт-Петербург: ГУАП, 2008. — 164 с.

27. *Кудинов Ю. И.* Основы современной информатики: учебное пособие / Ю. И. Кудинов, Ф.Ф. Пащенко. — Санкт-Петербург: Издательство "Лань", 2011. — 256 с.

28. *Плютто В. П.* Практикум по теории автоматического регулирования химико-технологических процессов / В. П. Плютто. — Москва: "Химия", 1969. — 112 с.

29. *Богатиков В. Н.* Моделирование систем автоматического регулирования технологических процессов: учебное пособие / В. Н. Богатиков, Л. В. Дранишников, А. Е. Пророков. — Петрозаводск: Издательство Петрозаводского государственного университета, 2011. — 20 с.

30. *Луценко В. А.* Аналоговые вычислительные машины в химии и химической технологии / В. А. Луценко, Л. Н. Финякин. — Москва: "Химия", 1979. — 244 с.

31. *Гуляев В. М.* Мониторинг окружающей среды / В. М. Гуляев, Л. В. Дранишников. — Днепропетровск: РВА "Дніпро-VAL", 2006. — 370 с.

Навчальне видання

Дранишников Леонід Васильович

**ІНТЕЛЕКТУАЛЬНІ МЕТОДИ
В УПРАВЛІННІ**

Навчальний посібник

Підписано до друку 06.12.2017. Формат 60×84 1/16.
Папір друк. Друк — різнограф. Ум.-друк. арк.24,18.
Тираж — 300. Зам. № 07/18.

Видавець і виготовлювач
Дніпровський державний технічний університет
51918, м. Кам'янське, вул. Дніпробудівська, 2

Свідоцтво про внесення суб'єкта видавничої справи
до державного реєстру видавництв серія ДК № 5399
від 26.07.2017 р.