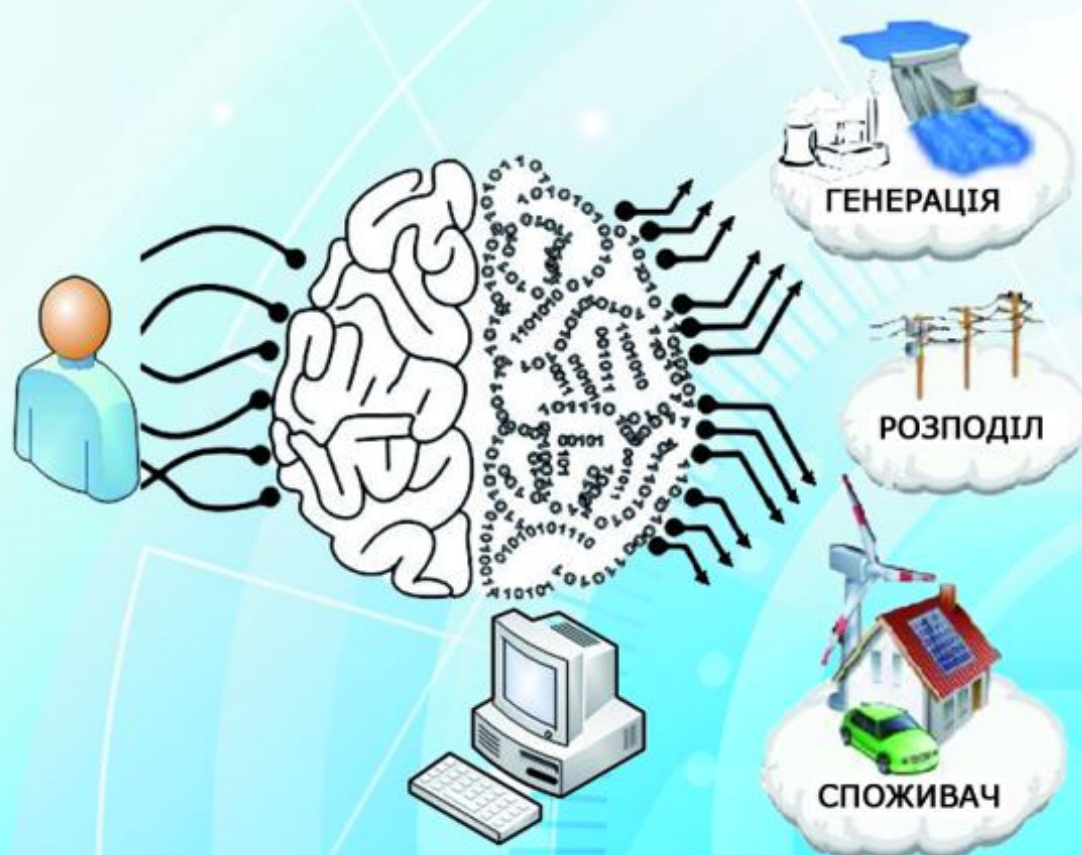


В. В. КИРИК

МАТЕМАТИЧНИЙ АПАРАТ ШТУЧНОГО ІНТЕЛЕКТУ В ЕЛЕКТРОЕНЕРГЕТИЧНИХ СИСТЕМАХ



Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

В. В. КИРИК

Математичний апарат штучного інтелекту в електроенергетичних системах

Підручник

*Затверджено Вченою радою КПІ ім. Ігоря Сікорського
як підручник для здобувачів ступеня магістра за спеціальністю
«Електроенергетика, електротехніка та електромеханіка»*

Київ
КПІ ім. Ігоря Сікорського
2019

Рецензенти:

В.В. Гавриленко, д-р фіз.-мат. наук, проф.,
Національний транспортний університет

М. С. Сегеда, д-р техн.наук, проф.,
Національний університет «Львівська політехніка»

**Відповідальний
редактор:**

М. В. Костерев, д-р техн.наук, проф.,
Національний технічний університет України
«Київський політехнічний інститут
імені Ігоря Сікорського»

Кирик В. В.

К43 Математичний апарат штучного інтелекту в електроенергетичних системах: підручник / В. В. Кирик.– Київ : КПІ ім. Ігоря Сікорського, Вид-во «Політехніка» 2019.– 224с.

ISBN 978-966-622-969-7

Розглянуто напрям створення інтелектуальних систем підтримки інженерних рішень для складних нелінійних електроенергетичних об'єктів з використанням математичного апарата штучного інтелекту. Висвітлено основні підходи, принципи створення, синтезу та налаштування нечітких логічних контролерів, штучних нейронних мереж і генетичних алгоритмів.

Для студентів усіх форм навчання факультетів електротехнічного та електронного напрямку дисципліни «Математичний апарат штучного інтелекту в електроенергетичних системах» та студентів-іноземців освітньо-професійної та освітньо-наукової програм підготовки «магістр» спеціальності 141 «Електроенергетика, електротехніка та електромеханіка» освітньої програми «Електричні системи і мережі», аспірантів, викладачів, а також інженерів електроенергетичних спеціальностей.

УДК 004.8:621.311](075.8)

ISBN 978-966-622-969-7

© В. В. Кирик, 2019

© КПІ ім. Ігоря Сікорського (ФЕА), 2019

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

- автоматичне введення резерву
- блок контролю управління
- виконавчий пристрій
- генетичний алгоритм
- електроенергетична система
- електронна обчислювальна машина
- електронна цифрова обчислювальна машина
- інформаційні системи
- інтелектуальний контролер
- коротке замикання
- лінія електропередавання
- Міжнародне Енергетичне Агентство
- мікроконтролер
- нечіткий модус поненс
- нечіткий модус толленс
- нечіткий пропорційно–інтегральний контролер
- нечітке асоціативне відображення
- нечітка логіка
- нечіткий логічний контролер
- нейронні мережі
- нечіткі нейронні мережі
- нечіткий пропорційно – інтегрально – диференціальний контролер
- пропорційний метод управління
- пропорційно–диференціальний метод управління
- пропорційно–інтегральний метод управління
- пропорційно – інтегрально – диференціальний метод управління
- трансформаторно–ключові виконавчі структури
- функція належності
- штучний інтелект
- широтно–імпульсна модуляція
- штучні нейронні мережі
- computer-aided software engineering
- IEEE Computational Intelligence Society
- Centre of Area, Bisector of Area (центр площі)
- Centre of Gravity (центр ваги або центроїд площі)
- Centre of Gravity for Singletons (центр ваги для одноточкових множин)

DMS/NMS/SCADA	– система оперативно–диспетчерського управління електричними мережами
FCL (MHY)	– Fuzzy Control Language (мова нечіткого управління)
FLC	– Fuzzy Logic Controller (нечіткий логічний контролер)
FMP	– fuzzy modus ponens (нечіткий модус поненс)
FMT	– fuzzy modus tollens (нечіткий модус толленс)
GIS	– геоінформаційна система
LM	– Left Most Maximum (ліве модальне значення)
OMS	– система управління аварійними вимиканнями
RM	– Right Most Maximum (праве модальне значення)
SAIDI	– індекс середньої тривалості вимикань
SAIFI	– індекс середньої частоти вимикань

ЗМІСТ

ВСТУП	8
РОЗДІЛ 1 ОСНОВНІ ПЕРЕДУМОВИ СТАНОВЛЕННЯ ІННОВАЦІЙНОЇ КОНЦЕПЦІЇ РОЗВИТКУ ЕЛЕКТРОЕНЕРГЕТИКИ	13
1.1 Проблеми класичних методів управління в електроенергетичних системах.....	13
1.2 Поняття, ідеологія та основні положення концепції інтелектуальних мереж у сучасній електроенергетичній системі.....	16
1.3 Тенденції розвитку інтелектуальних мереж у країнах світу.....	18
РОЗДІЛ 2 ШТУЧНИЙ ІНТЕЛЕКТ ТА МЕТОДИ І ТЕХНОЛОГІЇ ЙОГО СТВОРЕННЯ	23
2.1 Визначення штучного інтелекту та загальний опис проблеми його створення.....	23
2.2 Підходи до розуміння проблеми та методи створення систем штучного інтелекту	25
2.3 Напрями досліджень.....	28
2.4 Технології створення інформаційних систем з елементами штучного інтелекту.....	30
РОЗДІЛ 3 КЛАСИЧНІ МЕТОДИ УПРАВЛІННЯ В ЕЛЕКТРОЕНЕРГЕТИЧНИХ СИСТЕМАХ	35
3.1 Системи прийняття рішень щодо управління на основі класичних методів.....	35
РОЗДІЛ 4 МАТЕМАТИЧНИЙ АПАРАТ НЕЧІТКОЇ ЛОГІКИ ТА ЙОГО ВИКОРИСТАННЯ В ЕЛЕКТРОЕНЕРГЕТИЧНИХ ЗАДАЧАХ	46
4.1 Аналіз і характеристика підходів до створення систем з елементами штучного інтелекту.....	46
4.2 Особливості організації структурних схем інтелектуальних систем з нечітким логічним контролером.....	50
4.3 Методологія нечіткої логіки.....	56
4.4 Стохастична і лінгвістична невизначеності.....	61
4.5 Математичне визначення та характеристики чіткої і нечіткої множин...	65
4.6 Основні типи функцій належності.....	69
4.7 Операції перетину, об'єднання і вкладення нечітких множин.....	79
4.8 Математичні визначення нечіткої і лінгвістичної змінних.....	82
4.8.1 Поняття нечіткого висловлювання.....	83
4.8.2 Основні логічні операції з нечіткими висловлюваннями.....	85
4.9 Правила нечітких знань – продукцій.....	88

4.10 Прямий і зворотний методи нечіткого висновку.....	96
4.11 Основні етапи нечіткого логічного висновку.....	97
4.11.1 Формування бази правил систем нечіткого висновку.....	99
4.11.2 Фаззифікація (Fuzzification).....	100
4.11.3 Агрегування (Aggregation).....	102
4.11.4 Активація (Activation) – імплікація.....	103
4.11.5 Акумуляція (Accumulation).....	105
4.12 Дефаззифікація нечіткого логічного висновку.....	110
4.12.1 Метод центру ваги.....	111
4.12.2 Метод центру ваги для одноточкових множин.....	112
4.12.3 Метод центру площі.....	113
4.12.4 Метод лівого модального значення.....	113
4.12.5 Метод правого модального значення.....	114
4.13 Основні алгоритми нечіткого висновку.....	116
4.13.1 Алгоритм Мамдані (Mamdani).....	116
4.13.2 Алгоритм Цукамото (Tsukamoto).....	116
4.13.3 Алгоритм Ларсена (Larsen).....	117
4.13.4 Алгоритм Сугено (Sugeno).....	118
4.13.5 Спрощений алгоритм нечіткого висновку.....	118

РОЗДІЛ 5 ПРАКТИЧНА РЕАЛІЗАЦІЯ СИСТЕМИ З ЕЛЕМЕНТАМИ ШТУЧНОГО ІНТЕЛЕКТУ НА ОСНОВІ НЕЧІТКИХ ЛОГІЧНИХ КОНТРОЛЕРІВ.....

5.1 Загальні положення та особливості формування параметрів операцій за алгоритмом нечіткого контролера.....	142
5.2 Формування функцій належності, правил та налаштування нечіткого логічного контролера.....	150
5.3 Синтез та моделювання нечіткого логічного контролера з використанням програмного пакета FUZZY в середовищі MATLAB.....	154
5.3.1 Програмний пакет нечіткої логіки FUZZY в середовищі MATLAB та його особливості.....	155
5.3.2 Графічний інтерфейс редактора нечіткої системи висновку	159

РОЗДІЛ 6 НЕЧІТКИЙ КЛАСТЕРНИЙ АНАЛІЗ ДАНИХ.....

6.1 Загальна постановка задачі кластерного аналізу.....	169
6.2 Нечітка кластеризація і алгоритм її реалізації.....	171
6.3 Рішення завдання нечіткої кластеризації з використанням засобів графічного інтерфейсу в програмному додатку Fuzzy Logic Toolbox.....	174

РОЗДІЛ 7 ШТУЧНІ НЕЙРОННІ МЕРЕЖІ ТА ЇХ МОДЕЛІ.....

7.1 Поняття штучної нейронної мережі і основні способи її відтворення....	177
7.2 Математична модель штучного нейрона.....	179

7.3 Навчання штучної нейронної мережі.....	184
7.3.1 Алгоритм налаштування нейронної мережі.....	187
7.4 Загальна характеристика програмного інтерфейсу адаптивних систем нейронечіткого висновку.....	189

РОЗДІЛ 8 ОПТИМІЗАЦІЯ ПАРАМЕТРІВ

ЕЛЕКТРОЕНЕРГЕТИЧНИХ СИСТЕМ НА ОСНОВІ

МАТЕМАТИЧНОГО АПАРАТА ГЕНЕТИЧНИХ АЛГОРИТМІВ..... 191

8.1 Генетичний алгоритм та переваги його використання для оптимізаційних задач у порівнянні з класичними методами.....	191
8.2 Послідовність математичних операцій генетичного алгоритму.....	194
8.2.1 Формування початкової популяції.....	195
8.2.2 Селекція (відбір батьків).....	196
8.2.3 Розмноження (схрещування, кросовер, кросинговер).....	197
8.2.4 Мутація.....	198
8.3 Метод визначення оптимальної конфігурації трифідерної розподільної електричної мережі з використанням генетичного алгоритму.	200
8.3.1 Знаходження оптимальної конфігурації розподільної електричної мережі з використанням генетичного алгоритму.....	203
8.4 Навчальна програмна реалізація генетичного алгоритму.....	212

СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ..... 220

ПРЕДМЕТНИЙ ПОКАЗЧИК..... 223

ВСТУП

Протягом останніх трьох десятиліть дослідження в сфері штучного інтелекту (AI) дають змогу розробити широкий спектр методів і підходів, які можна адаптувати або використовувати для вирішення складних проблем електроенергетичної системи (наприклад, планування, експлуатація енергосистеми, управління режимами), які раніше вважалися нерозв'язними без суттєвих припущень в умовах невизначеності та відсутності достовірної інформації.

Концепція «розумної мережі», яка з'явилася на початку 21-го століття, завдяки перевагам інформаційних і комунікаційних онлайн технологій в електроенергетиці, дає змогу вирішити проблеми безпеки в централізованих енергосистемах, а також забезпечити збільшення інтеграції відновлюваних джерел енергії за рахунок розгортання сучасної вимірювальної інфраструктури в реальному часі для створення економічно привабливої ситуації як для споживачів, так і для постачальників електроенергії. Створення «розумної мережі» породило проблеми в різних сферах аналізу і проектування енергосистем, які не піддаються аналітичній обробці і мають бути вирішені з використанням інтелектуальних метаевристичних (навчальних) алгоритмів з галузі AI.

Хоча застосування інтелектуальних методів у деяких напрямках аналізу енергетичних систем досить складне (через їх обчислювальну складність), на сьогодні запропоновано широкий спектр методів AI, які націлені на модернізацію всього процесу електропостачання (що охоплює галузі виробництва, передавання, розподілу, споживання електроенергії, а також дерегулювання ринку електроенергії), які можна розділити на такі чотири основні класи:

1. Оптимізація систем відновлюваних і традиційних джерел енергії та формування гнучких електричних мереж – мікромереж; розподільних, районних та магістральних мереж; віртуальних електростанцій, вставок та ліній постійного струму; акумулюючих пристроїв електроенергії – на трьох рівнях: управління режимами, управління електроенергією на добу вперед і планування інвестицій;
2. Коротко– і довгострокове прогнозування попиту на електроенергію, цін на енергоносії та установлену потужність електроенергетичної системи;
3. Моніторинг, аналіз, оцінка стану і розпізнавання несправностей електроустаткування з властивостями самовідновлення функцій;
4. Забезпечення стійкості та надійності функціонування системи: субгармонічні низькочастотні коливання, кібербезпека, планетарні геомагнітні збурення.

Належна оптимізація має вирішальне значення для створення економічно ефективних схем проектування, експлуатації та управління. В цьому сенсі

математичні методи оптимізації традиційно використовувалися для надання аналітичних рішень з оптимізації класичних, нескладних гібридних систем. Однак зростання кількості змінних і обмежень, а також зростаюча нелінійність і невизначеності в системі, особливо у випадку складнозамкнених електричних мереж, призвели до того, що за допомогою точних методів стало неможливим вирішувати такі проблеми без ряду спрощуючих припущень, які значно занижують точність рішення. Поява генетичних алгоритмів оптимізації та нечіткої логіки і нейронних мереж відкриває можливості вирішення цієї проблеми.

Інтелектуальне прогнозування вхідних змінних для моделей енергосистеми може закласти основу для поліпшення цих моделей з метою кращого наближення до реальних ситуацій. У цьому контексті на сьогодні розроблено асортимент методів, які підвищують якість прогнозів щодо вихідних даних для численних моделей енергетичних систем. Залежно від тимчасового характеру проблеми (тобто горизонту прогнозування) аналіз прогнозу може бути виконаний протягом чотирьох прогнозованих часових горизонтів: ультракороткочасний (на кілька секунд наперед) динамічний аналіз продуктивності та стабільності системи; надкороткостроковий (від хвилин до годин) аналіз енергетичної безпеки, що дає змогу здійснювати профілактичний контроль системи; короткостроковий (у межах від години до 72 годин наперед) для економічної оцінки прийняття рішень щодо управління режимами, короткострокового планування технічного обслуговування і заявок на ринку енергоносіїв; довгостроковий (до декількох років наперед) для планування технічного обслуговування обладнання, моделювання енергетичної політики, планування розширення генеруючих потужностей і мереж та проектування електроенергетичних систем. До методів прогнозного моделювання, в контексті AI, можна віднести нечітку логіку та нейронні мережі.

Хоча вейвлет–перетворення традиційно застосовується для діагностики місць розташування та типів несправностей, які відбуваються в енергосистемах, розвиток аналітики структурування даних і AI виявив нові тенденції в цій галузі досліджень. У цьому контексті ряд методів інтелектуального аналізу даних, заснованих на розпізнаванні образів, було успішно застосовано для виявлення збоїв енергосистеми шляхом кластеризації та моделювання на основі нечітких множин і нечіткої логіки. Потенціал AI може використовуватися для забезпечення можливості самовідновлення енергосистем у критичних ситуаціях, що сприяє значному полегшенню здійснення автоматичного управління для підвищення надійності систем. Тому концепція мультиагентної системи, тобто метод розподіленого AI, який об'єднує кілька спільних агентів для виконання призначених завдань відповідно до загальної мети системи – є найбільш широко використовуваною стратегією.

На сьогодні час здійснено певні зусилля у напрямку використання підходів AI для створення інструментів з метою самовідновлення електричних

мереж: системи підтримки прийняття рішень на основі нечіткої логіки, штучних нейронних мереж, багатоцільових підходів на основі Парето.

Як майбутній напрям досліджень може бути вивчено ефективне застосування таких інструментів до інформації про ситуаційну обізнаність, отриману з інтелектуальних глобальних систем моніторингу або систем Інтернету речей (IoT), щоб забезпечити реалізацію цілей самовідновлення в системах масового енергопостачання під парадигмою смарт-мережі в доступному для огляду майбутньому.

Методи AI безумовно використовуються безпосередньо для вирішення складних проблем електроенергетичної системи і надалі відіграватимуть суттєву роль у модернізації всього електроенергетичного сектора від генерації до споживача, для забезпечення широкомасштабної інтеграції відновлюваних джерел в енергетичній мережі.

Предмет навчальної дисципліни базується на прищепленні знань у студентів з формування прийняття технічних рішень щодо підтримки функціонування електроенергетичних систем на основі сучасних комп'ютерно-інтегрованих технологій «штучного інтелекту».

Метою дисципліни є формування у студентів уявлень про алгоритми та процеси управління складними нелінійними системами в умовах непрогнозованих змін напруги електромережі живлення і невизначених змін нелінійності електроенергетичної системи на основі знань математичного апарата нечіткої логіки, нейронних мереж та генетичних алгоритмів.

Підготовка сучасного спеціаліста в галузі виробництва, передавання та розподілу електричної енергії неможлива без прищеплення йому навичок використання комп'ютеризованих інтелектуальних технологій розв'язання складних погано формалізованих електроенергетичних задач. Вивчення дисципліни «Математичний апарат штучного інтелекту в електроенергетичних системах» передбачає надання студентам уявлення про основні принципи побудови систем з елементами штучного інтелекту для підтримки рішень управління. Інженер-електрик, який спеціалізується в галузі передавання та розподілу електричної енергії, повинен орієнтуватися в передових інтелектуальних математичних методах аналізу процесів в енергетичних системах, які ґрунтуються на принципах штучного інтелекту, володіти базовими поняттями інтелектуальних систем, методами реалізації та технологіями побудови систем підтримки рішень. Під час вивчення дисципліни студент має отримати стійкі знання та навички, обсяг яких має бути достатнім для вирішення проектно-конструктських задач, що виникають під час проектування та впровадження технологій інформаційних інтелектуальних систем підтримки рішень.

Предмет навчальної дисципліни базується на прищепленні знань у студентів щодо використання математичного апарата штучного інтелекту для розв'язання практичних задач, зокрема в галузі електроенергетики, типової

структури системи та її основних функцій. Студенти набувають знань щодо формалізації методів та моделей розв'язання електроенергетичних задач у середовищі програм штучного інтелекту, зокрема, в програмних додатках FUZZY, ANFIS середовища MATLAB, програмних пакетах NeuroShell, GeneHunter; вивчають основні проблеми інженерії знань, пов'язані з використанням технологій інтелектуальних систем, основні моделі представлення знань у комп'ютерних системах та методи розв'язання практичних задач на їх основі. Набувають практичного досвіду щодо розв'язання задач логічного виведення в умовах неповноти або невизначеності вихідної інформації, знайомляться із основами нечіткої логіки, штучних нейронних мереж та генетичних алгоритмів.

Предмет навчальної дисципліни «Математичний апарат штучного інтелекту в електроенергетичних системах» зорієнтований на прищеплювання студентам спеціальності 141 «Електроенергетика, електротехніка та електромеханіка» освітньої програми «Електричні системи і мережі» уявлень про основні принципи побудови інтелектуальних систем підтримки рішень з елементами штучного інтелекту. Основні завдання навчальної дисципліни «Математичний апарат штучного інтелекту в електроенергетичних системах» подаються через систему знань, умінь і певного досвіду, наведених у цьому підручнику.

Задачами вивчення дисципліни «Математичний апарат штучного інтелекту в електроенергетичних системах» є глибоке оволодіння математичними знаннями для реалізації алгоритмів прийняття технічних рішень на основі нечіткої логіки, нейронних мереж та генетичних алгоритмів.

Практичне спрямування дисципліни «Математичний апарат штучного інтелекту в електроенергетичних системах» ґрунтується на реалізації вимог до підготовки кадрів, встановлених освітньо-кваліфікаційною характеристикою фахівця за спеціалізацією «Електричні системи і мережі», націлених на розв'язання основних передпроектних, технологічних, експлуатаційних, економічних і конструкторських задач, які виникають під час розрахунку, проектування і експлуатації електричних мереж і систем різних класів номінальної напруги.

Технологічне спрямування дисципліни «Математичний апарат штучного інтелекту в електроенергетичних системах» має озброїти студента сучасними математичними засобами аналізу і синтезу систем підтримки рішень щодо формування структур і топології електричних мереж, моделювання усталених і післяаварійних режимів, знаннями новітніх методів, засобів і способів формування керуючих впливів на режими роботи систем з метою вибору оптимальної стратегії управління режимами виробництва, передавання, перетворення і розподілу електричної енергії, можливістю розробки і впровадження в практику експлуатації засобів удосконалення цього процесу, дієвими засобами зниження технологічних втрат енергії в усіх ступенях ієрархії

електричної системи, засобами постановки експерименту в електричних мережах і системах, оцінювання похибки і вірогідності отриманих результатів, критичного їх осмислення й аналізу, прийняття оптимальних рекомендацій.

Інженерне спрямування дисципліни «Математичний апарат штучного інтелекту в електроенергетичних системах» покликане прищепити студенту уміння і навички роботи з сучасними інформаційними комп'ютерними системами для підтримки технічних рішень інженера–технолога, експлуатаційника, проектувальника і конструктора, які відповідають у повному обсязі їхнім виробничим функціям, тобто інженера, спроможного приймати самостійні, математично і логічно осмислені рішення під час проектування, конструювання і налагодження устаткування електричних мереж. Ці навички і уміння дають змогу молодому спеціалісту активно брати участь у рішенні проблем оптимального управління і регулювання режимів роботи електричних мереж та систем, у синтезі оптимальних схем побудови конфігурації електричної мережі, виборі оптимальних режимів роботи силового обладнання мережі та практичній роботі з ним тощо.

Підручник містить матеріали лекцій дисципліни «Математичний апарат штучного інтелекту в електроенергетичних системах» і складається з восьми розділів, що відображають п'ять змістовних модулів: «Інноваційні концепції розвитку електроенергетики», «Нечітка логіка та її використання для прийняття рішень в електроенергетичних системах», «Нечіткий кластерний аналіз даних», «Штучні нейронні мережі та їх моделі», «Генетичні алгоритми та оптимізація параметрів електроенергетичних систем».

Автор висловлює щиро подяку завідувачу кафедри інформаційних систем і технологій національного транспортного університету, д-ру фіз.–мат. наук, професору, В. В. Гавриленку, завідувачу кафедри електроенергетики та систем управління національного університету «Львівська політехніка», д-ру техн. наук, професору, М. С. Сегеді за рецензування підручника.

РОЗДІЛ 1

ОСНОВНІ ПЕРЕДУМОВИ СТАНОВЛЕННЯ ІННОВАЦІЙНОЇ КОНЦЕПЦІЇ РОЗВИТКУ ЕЛЕКТРОЕНЕРГЕТИКИ

1.1 Проблеми класичних методів управління в електроенергетичних системах

Електроенергетичні системи (ЕЕС) є нелінійними і багатопов'язаними системами, в яких протікають складні перехідні процеси і можливе виникнення критичних і хаотичних режимів. Проблема ефективного управління ЕЕС належить до числа фундаментальних науково–технічних проблем, гостроту і важливість якої підтверджують великі техногенні аварії, що сталися. Ефективне управління такими динамічними макросистемами є складним і не завжди адекватним для існуючих в енергетиці класичних методів автоматичного управління.

Традиційні алгоритми управління ЕЕС склалися понад за півстоліття тому і використовуються досі. Свого часу вони показали свою ефективність. Але їх застосування у зв'язку з підвищенням вимог до ЕЕС (стійкості, надійності, розширення функціональних можливостей) і логіка науково–технічного прогресу вимагають пошуку нових шляхів вдосконалення процесів управління енергооб'єктами і їх групами у складі ЕЕС.

Наведемо деякі приклади проблем у випадку застосування традиційних методів управління ЕЕС.

1. *Звуження сфери динамічної стійкості ЕЕС* пов'язане із застосуванням лінійних автоматичних регуляторів. Лінійні регулятори адекватно функціонують в заданій області відхилення від встановленого нормального стану енергооб'єкта. Наприклад, у пікових режимах турбогенератори працюють у режимах великих відхилень. Для ефективного управління, що забезпечує хоч би збереження асимптотичної (динамічної) стійкості енергосистеми, необхідно врахувати явища взаємопов'язаності і нелінійності процесів, що виникають в енергооб'єктах. Таким чином, параметри багатьох автоматичних регуляторів, розрахованих для конкретних режимів, не завжди забезпечують адекватне регулювання в критичних режимах і можуть навіть істотно погіршити статичні та динамічні властивості ЕЕС. Здолати вказаний недолік, пов'язаний з фіксованістю параметрів, наприклад лінійних систем автоматичного введення резерву (АВР), зокрема, для локальних режимів, можна, використовуючи методи нечіткої логіки.

2. *Робота в екстремальних режимах.* Як правило, це значні збурення режимів в ЕЕС, які спричиняються комутацією потужних навантажень, силових трансформаторів, ліній електропередавання (ЛЕП) і так далі. До ще складніших

змін режиму роботи ЕЕС призводять коротких замикань (КЗ), які залежно від виду можуть викликати критичні зміни потужності, що передається, різкі скидання потужності і, як результат, – можливість випадання генераторів з синхронізму. Наприклад, усунути втрату динамічної стійкості ЕЕС у разі трифазного КЗ можливо тільки шляхом швидкого відімкнення аварійної ділянки, що і є причиною різкого скидання потужності. Екстремальним режимом ЕЕС є також «лавина» частоти і напруги, що спричиняється зниженням генерованої потужності. Для унеможливлення таких режимів необхідно використовувати нелінійне взаємопов'язане управління частотою і потужністю генератора, що не завжди досягне у разі застосування традиційних пристроїв автоматичного регулювання. Зокрема, в цьому випадку можливе застосування синергетичної системи управління, яка формує у фазовому просторі енергооб'єкта динамічно стійкі аттрактори, на яких забезпечується стабілізація частоти і потужності відповідно до потужності навантажень. Синергетика основана на принципах нелінійної динаміки і є інтегральною наукою, яка включає такі принципи: нечіткої логіки, нейронних мереж, генетичних алгоритмів, теорії хаосу, кібернетики. Зміст та розуміння синергетики в тому, що у відкритих системах, які обмінюються з зовнішнім середовищем енергією або інформацією, виникають процеси стихійної самоорганізації, тобто зародження з фізичного хаосу деяких стійких упорядкованих структур зі зміненими властивостями системи.

3. *Розділення каналів управління енергооб'єктами.* Ця проблема актуальна за пікових і аварійних режимів ЕЕС. Наприклад, практично склалося так, що у разі застосування традиційних алгоритмів управління, підсистеми генератора і турбіни є розв'язаними, тобто локальними. Ще майже 40 років тому професор В. А. Веніков вказував на необхідність двоканалного взаємопов'язаного управління в подібних системах: «... потрібно одночасне узгодження управління збудженням турбогенератора і механічною потужністю його турбіни від одного комплексного управляючого пристрою». Більшість вживаних нині подібних систем управляються «по-старому». Тому виникають ситуації, коли дії каналів управління підсистем генератора і турбіни вступають в явні протиріччя один з одним, що може істотно знизити стійкість ЕЕС. Зокрема, для вирішення вказаної проблеми найбільш підходять інтелектуальні методи управління на штучних нейронних мережах (ШНМ). У цьому випадку кількість каналів управління може бути значно більшою двох. Це дасть змогу задіяти різні контури управління і стабілізації, сигнали яких пропорційні режимам енергооб'єктів, що живляться від цього генератора.

4. *Виникнення системних аварій.* У ЕЕС можливі збурення, що призводять до явища електромеханічного резонансу, які здатні призвести до порушення стійкості ЕЕС, асинхронного ходу і розвитку системної аварії. Це обумовлює необхідність вирішення такої проблеми у разі побудови автоматичних регуляторів для генераторів, що забезпечують зменшення

найважчих збурень з боку ЕЕС. Проблема ефективно вирішується шляхом синтезу синергетичних алгоритмів управління енергооб'єктами за їх якнайповнішими нелінійними моделями, що припускає застосування методів нечіткої логіки.

5. *Параметрична невизначеність.* Практично будь-яка ЕЕС характеризується наявністю ряду невизначеностей: неточним описом математичної моделі, не до кінця контрольованою зміною статичних і динамічних параметрів, дією на систему зовнішніх збурень тощо. Наявність різного роду невизначеностей зумовила розвиток адаптивних систем управління. Незважаючи на це, прагнуть як і раніше використати лінійну апроксимацію (лінеаризацію) і методи, ґрунтовані на цьому підході: модальне управління, локально-оптимальне управління, метод швидкісного градієнта та ін. Параметрична невизначеність також вирішується застосуванням синергетичних алгоритмів управління енергооб'єктами – застосуванням нечітких нейронних мереж (ННМ) і методів нечіткої логіки.

На ряду вирішення згаданих проблем необхідно враховувати такі особливості ЕЕС:

а) структурної складності ЕЕС, що виявляється у багатовимірності, багатопов'язаності і параметричній невизначеності; б) багаторежимності функціонування, пов'язаній з великою зміною навантажень та істотною зміною конфігурації мережі після вимкнення частини аварійних ділянок; в) прояви в перехідних і особливо екстремальних режимах нелінійних властивостей об'єктів ЕЕС, що призводять до істотних похибок розрахунків і якісних спотворень результатів під час спроби застосування методів лінеаризації подібних режимів.

Найбільш відповідним варіантом вирішення вказаних проблем з урахуванням особливостей процесів в ЕЕС є використання інтелектуальних технологій управління та застосування принципів штучного інтелекту (ШІ, *англ.* AI) – науки, що забезпечує вирішення проблем управління складними процесами самоорганізації і колективної, когерентної поведінки в нелінійних динамічних системах різної природи з відтворенням дій людини мікропроцесорними обчислювальними засобами.

Сенс і зміст використання ШІ в електроенергетичних системах полягає в забезпеченні оптимального рівня перетворення енергії та обміну при цьому інформацією між складовими елементами системи на основі інтелектуальних методів. Основою математичного апарату штучного інтелекту є нейронні мережі, нечітка логіка, генетичні алгоритми, теорія хаосу та кібернетика.

1.2 Поняття, ідеологія та основні положення концепції інтелектуальних мереж у сучасній електроенергетичній системі

Сучасна логіка науково-технічного прогресу та розвитку суспільства спонукають до необхідності розроблення нової концепції інноваційного розвитку електроенергетики на основі розумних мереж з елементами штучного інтелекту, яка, з одного боку, відповідає б сучасним поглядам, цілям і цінностям економічного соціального і суспільного розвитку, а з іншого – максимально враховувала основні тенденції та напрямки науково-технічного прогресу в усіх галузях, сферах життя та діяльності суспільства.

Інтелектуальна мережа – це автоматизована система, яка самостійно має відстежувати і розподіляти потоки електроенергії для досягнення максимальної ефективності використання енергетичних ресурсів. Це повністю автоматизована енергетична система, що забезпечує двосторонній потік електричної енергії та інформації між електричними станціями і споживачами за рахунок застосування новітніх технологій, інструментів і методів, наповнює електроенергетику «знаннями», тобто елементами штучного інтелекту, які в цілому дають змогу різко підвищити ефективність функціонування електроенергетичної системи.

Інтелектуальні мережі передбачають необхідність організаційних змін, нової моделі процесів, нових математичних рішень у галузі інформаційних технологій, а також сучасних рішень у області автоматизованих систем управління технологічними процесами і диспетчерського управління в електроенергетиці.

Ключові функції системи *Інтелектуальних мереж* такі:

1. Діагностичне технічне обслуговування та ремонт / Система управління для підвищення рентабельності (Condition Based Predictive Maintenance / Asset Management);
2. Аналіз реагування на попит (Demand Response Analysis);
3. Пряме управління навантаженням у споживача (Direct Customer Load Control);
4. Моделювання диспетчеризації (Dispatcher Training and Simulation);
5. Інструменти аналізу розподіленої системи в режимі реального часу (Distribution System Real-Time Analysis Tools);
6. Підтримка аварійного відновлення системи (Emergency System Restoration Support);
7. Виявлення несправностей в автоматичному режимі (Fault Detection);
8. Обладнання моніторингу передавання електроенергії (Feeder Equipment Monitoring);
9. Інтегрована система контролю напруги (Integrated Volt VAR Control);
10. Інтеграція систем різних рівнів і функціоналу, включаючи DMS (OMS), AMI, GIS (Integration of DMS (OMS) AML GIS):

(DMS/NMS/SCADA) – Система оперативно–диспетчерського управління електричними мережами;
 (OMS) – Система управління аварійними вимиканнями;
 (GIS) – Геоінформаційна система;

11. Прогнозування навантаження (Load Forecasting);
12. Багаторівнева конфігурація пристроїв передавання електроенергії (Multilevel Feeder Configuration);
13. Управління, аналіз і оптимізація рішень щодо мережевої комутації (Networks Mtching Management Analysis Optimization);
14. Оцінка якості електроенергії (Power Quality Assessments);
15. Управління релейним захистом (Relay Protection Coordnation).

Журнал Transmission & Distribution World та консалтингова компанія Black & Veatch провели глобальне дослідження, спрямоване на оцінювання спрямованості електромережових компаній в галузі розвитку Smart Grid (Black & Veatch і T & D World), опитали представників 91 компанії (рис.1.1).



1– Системи автоматизованого обліку; 2 –Телекомунікації; 3 – Автоматизація розподільних мереж; 4 – Автоматизація підстанцій; 5 – Управління і моніторинг навантаження; 6 – Моніторинг і контроль стану мережі; 7 – Управління мобільними бригадами; 8 – Система управління мережами; 9 – Системи управління бізнес–процесами; 10 – Управління навантаженням; 11 – Розподілена генерація; 12 – Системи «розумний будинок»; 13 – Системи FACTS та ін.

Рис. 1.1. Кількість респондентів з інвестиціями в напрямки розвитку інтелектуальних мереж

Близько 80 % респондентів планують реалізовувати проекти із створення інтелектуальної мережі, вважаючи підвищення надійності головним чинником, що обумовлює актуальність цього завдання, і зниження операційних витрат другого за значенням фактора. Близько 23 % респондентів зазначили, що вони планують витратити на реалізацію проектів Smart Grid від п'яти до 10 млн. дол, 21 % з них підтвердили обсяг інвестицій від 1 до 5 млн. дол. Переважна більшість опитаних (79 %) назвали розвиток систем автоматизованого комерційного обліку енергоресурсів першочерговим завданням на шляху до Smart Grid. Також відзначається висока позиція

систем автоматизації розподільних мереж і систем управління та моніторингу навантаження в рейтингу технологій інтелектуальних мереж. Результати дослідження представлено на рис.1.1.

1.3 Тенденції розвитку інтелектуальних мереж у країнах світу

Електроенергетична система є середовищем, в якому виникають проблеми, і сама по собі не є статистичною цілісністю. Система динамічна, оскільки завжди змінюється, піддаючись як зовнішнім, так і внутрішнім впливам. Енергосистема змінюється разом зі своїми проблемами і їх рішеннями у фізичному і концептуальному просторі. В результаті технологічного процесу також змінюються і фільтруються зв'язки між елементами системи.

Електроенергетична система є досить складною і представляє собою сукупність величезного числа різних об'єктів, що діють разом. Складністю є взаємодія і, більше того, взаємозалежність, тобто поведінка одного або декількох елементів впливає на поведінку інших.

Оптимальне рішення щодо управління для множини вирішуваних завдань або їх набору як цілого не може бути отримано розчленовуванням на окремі проблеми. Складні проблеми управління в електроенергетиці мають вирішуватися не ізольовано на основі односторонніх причинних відносин. Складність вирішуваних задач в електроенергетиці формує картину взаємовідношення і множинної причинності. Точний математичний опис взаємодії в системі виконати досить складно: занадто часто взаємозв'язок проблем виявляється тільки після того, як ми застосували «рішення щодо управління», які породжують вторинні проблеми.

Сьогодні світовий попит на електроенергію зростає швидше, ніж попит на первинні енергоносії. Згідно з прогнозом Міжнародного Енергетичного Агентства (МЕА) до 2030 р. темпи зростання попиту на електроенергію випереджатимуть у 1,5–2 рази темпи зростання попиту на первинні енергоносії. Найвищий темп зростання попиту на електроенергію спостерігається в країнах Азії.

Дефіцит резервів потужності генерації за періоди пікових навантажень, особливо в умовах інтенсивного нарощування потужностей, може призводити до розбалансування режиму роботи енергосистеми і як наслідок – до порушень сталого електропостачання. Дослідження, проведені в Європейському Союзі, показали, що намітилися тенденції зниження резервів генерації практично у всіх європейських країнах.

Діючі сьогодні енергосистеми побудовано за схемою «централізованого енергопостачання», яка потребує використання більш високої напруги і створення великомасштабних електромереж. У мережах такого типу локальні збої можуть впливати на всю енергосистему і часто призводити до великих аварій.

До числа найбільш істотних змін у розвитку суспільства та економіки, які впливають на енергетичну галузь, зарубіжними дослідженнями віднесено:

1. Дефіцит джерел електричної енергії.
2. Постійно зростаючі вимоги до надійності та якості електропостачання з боку споживачів.
3. Постійне підвищення вартості електричної енергії у всьому світі.
4. Старіння і зростаючий дефіцит кваліфікованих кадрів у енергетичній галузі.
5. Зростання вимог заінтересованих сторін – споживачів до результатів діяльності енергетичних компаній. Зміна організаційних форм власності та формування ринкових умов зумовили виникнення для енергетичних компаній нової системи вимог стейкхолдерів (акціонерів, законодавців, регулюючих органів, споживачів, громадських та екологічних організацій), суть яких полягає у підвищенні надійності електропостачання, зниженні операційних витрат, підвищенні доходів інвесторів, зниженні чисельності персоналу тощо.
6. Вимоги екологічної та промислової безпеки функціонування енергетичних об'єктів.
7. Зниження загальносистемних витрат.

Необхідність створення нової концепції розвитку електроенергетики була продиктована економічним зростанням, нерозривно пов'язаним зі збільшенням обсягу енергоспоживання і підвищенням вимог до якості та рівня надійності енергопостачання. У результаті проведеного відповідними аналітичними відомствами США і ЄС аналізу стану та перспектив розвитку світової енергетичної сфери зроблено висновок, що успішне розв'язання нових проблем у рамках колишньої концепції екстенсивного розвитку електроенергетики переважно шляхом лише нарощування потужностей і розширення кількісного складу енергетичного та електротехнічного обладнання, навіть з поліпшеними характеристиками, виявляється недостатнім. Тому в більшості розвинених зарубіжних країн все більше уваги приділяється питанням впровадження в електроенергетиці інформаційних технологій на основі штучного інтелекту з реалізацією розумних мереж (Smart Grid) як основи майбутнього розвитку енергетики, основаної на таких вихідних положеннях:

1. Системна модернізація галузі охоплює всі її складові: генерацію електроенергії, передавання і розподіл, системи обліку та збут, диспетчеризацію і управління енергоспоживанням.
2. Електрична мережа (усі її сегменти) розглядається як основний об'єкт формування нового технологічного базису, розвитку функціональних властивостей енергосистеми.

3. Енергетична система розвивається як «інтернет-подібна» інфраструктура з формуванням в енергетичній, інформаційній, економічній і фінансовій сферах взаємовідносин між усіма суб'єктами енергетичного ринку та іншими заінтересованими сторонами.
4. Процес формування концепцій охоплює весь комплекс робіт – від попередніх досліджень до широкого впровадження інновацій на всіх рівнях інноваційного розвитку електроенергетики – нормативно-правовому, технологічному, технічному, організаційному, управлінському та інформаційному.
5. Розроблення та реалізація концепцій та відповідних програм впровадження «інтелектуальних» технологій носить інноваційний характер і дає поштовх до переходу на новий технологічний уклад в електроенергетиці та економіці в цілому.

Нові умови і каталізатори розвитку галузі формують потребу в розробленні та впровадженні нових технологій і елементів на основі штучного інтелекту, які забезпечують: рух потоків електроенергії та інформації від енергетичних компаній до споживачів і в зворотному напрямку; постійний контроль за всіма елементами мережі від роботи електростанцій до споживання електроенергії індивідуальними пристроями; інтеграцію розподілених джерел електроенергії (у тому числі поновлюваних) і засобів зберігання електроенергії; рекуперацію тепла тощо (рис. 1.2).

Smart Grid – це автоматизована система, яка самостійно відстежує і розподіляє потоки електроенергії для досягнення максимальної ефективності використання енергетичних ресурсів. Порівняння функцій класичної (традиційної) мережі та розумної на базі концепції інтелектуальних технологій представлено в табл.1.1.



Рис.1.2. Концепція інтелектуальних мереж

Табл.1.1. Порівняння функцій енергетичних систем

Класична електроенергетична система	Електроенергетична система на базі концепції інтелектуальних технологій
Одностороння комунікація між елементами або її відсутність	Двосторонні комунікації
Централізована генерація – розподілена генерація із складним процесом інтеграції	Централізована та розподілена генерація оптимально адаптовані в мережу
Топологія – переважно радіальна, кільцева	Переважно складнозамкнена
Реакція на наслідки аварії	Реакція на передбачення та попередження аварії
Робота обладнання до відмови	Моніторинг і самодіагностика, які продовжують функціонування обладнання
Ручне відновлення	Автоматичне відновлення – «самоорганізовані мережі»
Схильність до системних аварій	Передбачення розвитку системних аварій
Ручне і фіксоване виділення мережі	Адаптивне виділення. Управління в реальному часі
Перевірка обладнання на місці	Віддалений моніторинг обладнання
Обмежений контроль перетікання потужності	Управління перетіканням потужності. Навантаження адаптується до генерації
Недоступна або надто запізнена інформація про ціну для споживача	Ціна в реальному часі

КОНТРОЛЬНІ ЗАПИТАННЯ ТА ЗАВДАННЯ

1. Наведіть приклади проблем застосування традиційних методів управління ЕЕС.
2. Який характер мають передатні характеристики класичних систем прийняття рішень щодо управління?
3. Які особливості ЕЕС необхідно враховувати приймаючи рішення щодо управління.
4. Обґрунтуйте сенс і зміст синергетики.
5. Дайте визначення поняттю Інтелектуальна мережа.
6. Ключові функції системи Інтелектуальних мереж.
7. Тенденції розвитку інтелектуальних мереж у країнах світу.
8. Порівняйте функції класичної та «Smart Grid» електроенергетичних систем.

ЛІТЕРАТУРА

[13], стор. 15–18; [21], стор. 12–20; [11], стор. 7–9; [4], стор. 5–22; [17], стор. 6–20; [27], стор. 2–26; [30], стор. 2–18.

РОЗДІЛ 2

ШТУЧНИЙ ІНТЕЛЕКТ ТА МЕТОДИ І ТЕХНОЛОГІЇ ЙОГО СТВОРЕННЯ

2.1 Визначення штучного інтелекту та загальний опис проблеми його створення

Інтелект (intellectus, сприйняття, відчуття, розуміння) – це сукупність розумових здібностей, тобто інформаційний потенціал знань конкретної особистості, отриманий у результаті функціонування свідомості, мислення та розуму людини.

Людина застосовує інтелект для обробки *наявної інформації*, наприклад, з метою побудови або вдосконалення розуміння, позиції, стратегії, методу, правила, комбінації, відношення, пояснення, рішення, плану чи цілі.

Інтелект пов'язаний з іншими внутрішніми властивостями людини, такими як сприйняття, пам'ять, мова, уява, самосвідомість, самоконтроль, характер, володіння тілом, творчість, інтуїція і власне формується завдяки функціонуванню означених параметрів особистості.

Штучний інтелект (Artificial intelligence, AI) – розділ інформатики, що забезпечує вирішення проблем та завдань, які відтворюють дії, виконувані людиною, з використанням мікропроцесорних обчислювальних засобів, алгоритми роботи яких формуються на основі сучасних математичних методів. Тобто штучний інтелект (за визначенням професора Буткевича О. Ф.) – це комплексний апарат автоматизованого пошуку на базі обчислювальних засобів розв'язків задач управління складними системами через дії з їх математичними моделями для досягнення максимуму заданих критеріїв оптимальності.

Штучний інтелект – це вершина, до якої прямує розвиток інформаційних технологій.

Зауваження

Штучний інтелект – дуже молода галузь досліджень, започаткована 1956 року. Її історичний шлях нагадує синусоїду, кожен «зліт» якої ініціювався деякою новою ідеєю. На сьогодні її розвиток перебуває на «спаді» і поступається застосуванню вже досягнутих результатів у інших галузях науки, промисловості, бізнесі та навіть повсякденному житті.

Штучний Інтелект – це не тільки Розум, це не тільки пізнання або навчання. Люди, які розробили роботів Deep Blue і AlphaGo, були завбачливими в тому, що працювали разом як команда, аби вирішити проблему, яку ніхто не міг вирішити раніше.

У людському інтелекті є ще така складова, як мудрість – бути хорошою людиною, яка допомагає іншим, коли це можливо, вносить свій внесок до спільноти і працює з іншими, щоб зробити світ кращим. Люди мудріші, оскільки працюють разом, поважають один одного і допомагають один одному в суспільстві. Вони мудріші, оскільки роблять правильні речі, наприклад, доглядають за тваринами, які не можуть доглядати за собою.

Роботи зі штучним інтелектом ще не такі розвинені, як люди, в співпраці з іншими людьми або навіть у співпраці з іншими роботами. Для роботів командна робота є надзвичайно складним завданням. Це одна із проблем створення штучного інтелекту.

Незважаючи на те, що сьогодні у нас є досить розумні роботи, які роблять багато речей краще, ніж люди, але вони ще не мають такого інтелектуального потенціалу, як люди. Скільки часу знадобиться роботам, щоб стати по-справжньому розумнішими за людей? Це дуже складно передбачити.

Люди креативні, і будь-який науковець може дуже скоро запропонувати розумну ідею для створення цих суперрозумних роботів.

Визначення 1. З боку інформаційних систем: Інтелектуальною називається система, здатна цілеспрямовано залежно від стану інформаційних входів змінювати не тільки параметри функціонування, але і сам спосіб своєї поведінки, причому спосіб поведінки залежить не тільки від поточного стану інформаційних входів, але також і від попередніх станів системи.

Визначення 2. З боку обчислювальних систем: Інтелектуальною називається система, що моделює на комп'ютері мислення людини. Друге визначення з'явилося в 60-х роках, коли вважалося, що мозок людини можна змоделювати на комп'ютері. Клітини мозку – нейрони – програмно описувалися спеціальними математичними методами.

Визначення 3. З боку автоматизованих систем: Інтелектуальною називається система, що дає змогу посилити інтелектуальну діяльність людини за рахунок ведення з ним осмисленого діалогу. Необхідно, щоб комп'ютер служив порадиником людині, швидко аналізував ситуацію, генерував варіанти дії на основі знань величезної пам'яті і пропонував їх людині, а людина розглядала запропоновані варіанти і аналізувала, чому той чи інший варіант є прийнятним або ні. Комп'ютер, з огляду на отримані роз'яснення, знову аналізував би всі варіанти дії і видавав нові, а людина вибирала б відповідний варіант і несла відповідальність за його реалізацію.

Точного визначення штучного інтелекту немає, оскільки у філософії не розв'язано питання про природу і статус людського інтелекту. Немає і точного критерію досягнення комп'ютером «розумності», хоча перед штучним інтелектом було запропоновано низку гіпотез, наприклад, тест Тюрінга або гіпотеза Ньюела–Саймона. Нині існує багато підходів як до розуміння задач штучного інтелекту, так і до створення інтелектуальних систем.

Одна з класифікацій виділяє два підходи до створення систем штучного інтелекту:

низхідний, семіотичний – створення символічних систем (лінгвістичних), що моделюють високорівневі психічні процеси: мислення, судження, мову, емоції, творчість тощо;

висхідний, біологічний – створення нечітких логічних контролерів, штучних нейронних мереж і алгоритмів еволюційних обчислень, що моделюють інтелектуальну поведінку на основі менших «неінтелектуальних» елементів.

Як і всі комп'ютерні науки, штучний інтелект використовує математичний апарат. Особливе значення для неї мають філософія, лінгвістика і робототехніка.

2.2 Підходи до розуміння проблеми та методи створення систем штучного інтелекту

На сьогодні не існує єдиної відповіді на питання, чим опікується штучний інтелект (ШІ). В основному наука ШІ зводиться до такого:

- штучний інтелект спрямований на розв'язання завдань, які потребують людського розуміння. Отже, йдеться про те, щоб навчити ШІ розв'язувати тести інтелекту. Це передбачає розвиток способів розв'язання задач за аналогією, методів дедукції та індукції, накопичення базових знань і вміння їх використовувати;
- штучний інтелект спрямований на розв'язання задач, для яких не існує способів розв'язання або вони не коректні (через обмеження в часі, пам'яті тощо). Штучний інтелект займається моделюванням людської вищої нервової діяльності;
- штучний інтелект – це системи, які можуть оперувати зі знаннями, а найголовніше – навчатися;
- штучний інтелект (почав розвиватися з 1990-х років) – це алгоритми пошуку і прийняття рішення з агентно-орієнтованим напрямком, який зосереджує увагу на тих методах і алгоритмах, що допомагають інтелектуальному агенту виживати в довкіллі під час виконання свого завдання.

Найзагальніший підхід полягає в тому, що штучний інтелект матиме змогу поводити себе як людський за звичних умов. Ця ідея є узагальненим підходом тесту Тюрінга, який стверджує, що машина стане розумною тоді, коли буде спроможна підтримувати діалог зі звичайною людиною, а та не зможе зрозуміти, що розмовляє з машиною (діалог відбувається переписуванням).

Часто пропонують ще один підхід: штучний інтелект виникає тоді, коли машина буде відчувати і творити, будучи спроможною до спілкування та навчання, і мати змогу отримати емоції та інтуїцію.

Серед негативних аспектів штучного інтелекту можна відмітити те, що сучасні інформаційні технології полегшують здійснення не тільки корисних справ, але й різних зловживань, злочинів та ін.

За допомогою спеціального програмного середовища мікропроцесорна система може реалізувати певні ознаки штучного інтелекту. Мова йде про створення апаратно-програмної системи, яка буде здатна реалізувати певні

моделі інтелекту. Інтелектуальною слід вважати систему, яка вміє орієнтуватися і планувати свою поведінку в новій ситуації.

Серед методів створення систем штучного інтелекту можна виділити основні чотири: *логічний, структурний, еволюційний, імітаційний*.

1. *Логічний*. Основою логічного методу є алгебра логіки. Розвиток алгебра логіки набула у вигляді математичних операцій з предикатами, в якій вона розширена за рахунок введення предметних символів, відношень між ними. Кожна така система має блок генерації цілі, і блок виведення намагається довести таку ціль як теорему. Якщо ціль досягнута, то послідовність використаних правил дасть змогу отримати ланцюжок дій, необхідних для реалізації поставленої цілі (таку систему ще називають експертною системою). Потужність такої системи визначається можливостями генератора цілей і цифрового доведення теорем. Для досягнення кращої виразності логічний метод використовує новий напрям, назва його – нечітка логіка. Головною відмінністю цього напрямку є те, що істинність вислову може набувати окрім значень «так»/«ні» (1/0) ще й проміжних значень – «не знаю» (0,5), «струм швидше сильний, ніж слабкий» (0,75), «струм швидше слабкий, ніж сильний» (0,25). Такий підхід подібніший до мислення людини, оскільки вона рідко відповідає «так» або «ні».
2. *Структурний*. Під структурним методом ми розуміємо спроби побудови ШІ шляхом моделювання нервової структури людського мозку. Однією з перших таких спроб був перцептрон Френка Розенблатта, реалізований у вигляді електронної машини «Марк-1».

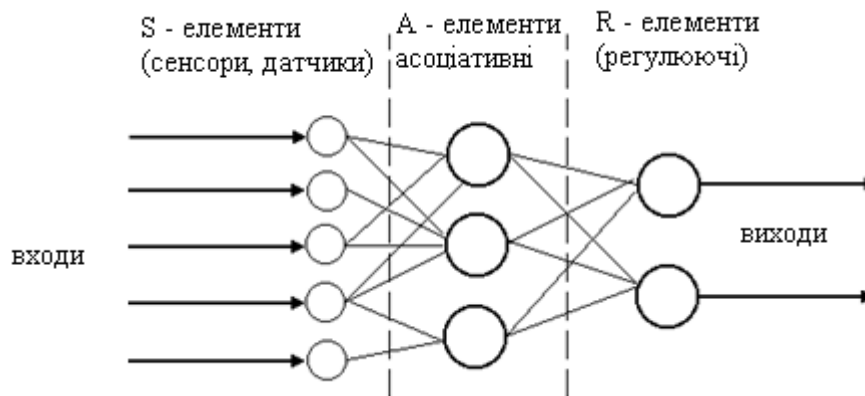
Перцептрон Розенблатта передавав повідомлення від «ока», яке реалізовувалось системою фотоелементів, в блоки електромеханічних комірок пам'яті, які оцінювали відносні величини електричних сигналів. Ці комірки з'єднувались між собою випадковим чином, створюючи мережу з прямими зв'язками. Зазначимо, що в ній були відсутні зворотні зв'язки між нейроноподібними елементами. Перцептрон міг навчатись шляхом спроб і помилок, а також корекцією електричних імпульсів.

Головною моделюючою структурною одиницею в перцептронах (як і в більшості інших варіантів моделювання мозку) є нейрон. Пізніше виникли й інші моделі, відоміші під назвою нейронні мережі (НМ) і їхні реалізації – нейрокомп'ютери. Ці моделі відрізняються за будовою окремих нейронів, топологією зв'язків між ними і алгоритмами навчання. Серед найвідоміших на початку 2000-х років варіантів НМ можна назвати НМ зі зворотним розповсюдженням помилки, сітки Кохонена, сітки Хопфілда, стохастичні нейрони сітки. У ширшому розумінні цей підхід відомий як *конективізм* (використання штучних нейронних мереж для моделювання когнітивних – розумових процесів розвитку). Відмінності між логічним та структурним підходами не стільки

принципові, як це здається на перший погляд. Алгоритми спрощення і вербалізації (словесного вираження) нейронних мереж перетворюють моделі структурного підходу на явні логічні моделі. З іншого боку, ще 1943 року Маккалок і Піттс показали, що нейронна сітка може реалізувати будь-яку функцію алгебри логіки.

Зауваження

Найпростіший перцептрон складається з трьох шарів.



На першому шарі розташовані чутливі рецепторні елементи (S – елементи), на які подаються вхідні зображення. S – елементи пов’язані з елементами другого шару (А – елементами). А – елемент збуджується, тобто генерує вихідний сигнал, тоді і тільки тоді, коли збуджується достатня кількість зв’язаних з ним S – елементів.

3. *Еволюційний*. Під час побудови системи ШІ за цим методом основну увагу зосереджують на побудові початкової моделі та правил, за якими вона може змінюватися (еволюціонувати). Причому модель може бути створено за найрізноманітнішими методами. Це може бути і НМ, і набір логічних правил, і будь-яка інша модель. Завдяки обчислювальному алгоритму комп’ютер на основі перевірки моделей відбирає найкращі з них, і за цими моделями за найрізноманітнішими правилами генеруються нові моделі. Серед еволюційних алгоритмів класичним вважається генетичний алгоритм.
4. *Імітаційний*. Цей метод є класичним для кібернетики і порівнюється з одним із її базових понять «чорний ящик». Об’єкт, поведінка якого імітується, саме і є «чорним ящиком». Для нас не важливо, які моделі у нього всередині і як він діє, головне, щоб наша модель в аналогічних ситуаціях поводи́ла себе без змін. Цим методом моделюється інша властивість людини – здатність копіювати те, що роблять інші, без поділу на елементарні операції і формального опису дій. Часто ця властивість заощаджує багато часу об’єктові, особливо на початку його життя.

У гібридних інтелектуальних системах об'єднують ці методи. Експертні правила висновків можуть генеруватися нейронними мережами, а побіжні отримують за допомогою статистичного вивчення. Перспективний новий підхід, який ще називають підсиленням інтелекту, розглядає досягнення ШІ у процесі еволюційної розробки як поточний ефект підсилення людського інтелекту технологіями.

2.3 Напрями досліджень

У випадку створення ШІ можна виділити такий вагомий напрям, як моделювання міркувань. Тривалий час розвиток цієї науки просувався саме таким шляхом, і зараз це одна з найрозвиненіших сфер у сучасному ШІ. Моделювання міркувань має на увазі створення символічних систем, на вході яких поставлена деяка задача, а на виході очікується її розв'язок. Як правило, запропонована задача уже формалізована, тобто переведена на математичну форму, але або не має алгоритму розв'язання, або цей алгоритм занадто складний, трудомісткий тощо. До цього напрямку входять: доведення теорем, прийняття рішень і теорія ігор, планування і диспетчеризація, прогнозування.

Так, на перший план виходить інженерія знань, яка об'єднує завдання отримання знань з простої інформації, їх систематизацію і використання. Досягнення в цій сфері торкаються майже всіх інших напрямків дослідження ШІ. Тут також необхідно відзначити дві важливі підобласті. Перша з них – *машинне навчання* – стосується процесу *самостійного* отримання знань інтелектуальною системою під час її роботи. Друга пов'язана зі створенням *експертних систем* – програм, які використовують спеціалізовані бази знань для отримання достовірних висновків щодо довільної проблеми.

Великі та цікаві досягнення є в сфері моделювання біологічних систем, до яких можна віднести кілька незалежних напрямків. Нейронні мережі використовуються для розв'язання нечітких і складних проблем, таких як розпізнавання геометричних фігур чи кластеризація об'єктів. *Генетичний підхід* заснований на ідеї, що деякий алгоритм може стати ефективнішим, якщо відбере найкращі характеристики у інших алгоритмів («батьків»). Відносно новий підхід, де ставиться задача створення автономної програми – агента, котрий співпрацює з довкіллям, називається *агентний підхід*. А якщо належним чином змусити велику кількість «не дуже інтелектуальних» агентів співпрацювати разом, то можна отримати *«мурашиний» інтелект*.

Задачі розпізнавання об'єктів вже частково розв'язуються в рамках інших напрямків, а саме розпізнавання символів, рукописного тексту, мови, аналіз текстів. Особливо слід згадати *комп'ютерне розпізнавання образів*, яке пов'язане з машинним навчанням та робототехнікою.

Робототехніка і штучний інтелект часто поєднуються одне з одним. Об'єднання цих двох наук, створення інтелектуальних роботів, можна вважати ще одним напрямком ШІ.

Окремо тримається *машинна творчість (Computational creativity)*, через те, що природа людської творчості ще менше вивчена, ніж природа інтелекту. Проте ця галузь існує, і тут постають проблеми написання комп'ютером програм свого ж функціонування, щоб комп'ютери стали здатні генерувати нові рівні розвитку ШІ без найменшої допомоги з боку людей.

Як приклади, можна навести програмування інтелекту в комп'ютерних іграх, нелінійному керуванні, інтелектуальні системи безпеки.

Машинне навчання – це розділ штучного інтелекту, що має за основу побудову та дослідження систем, які можуть самостійно навчатись з даних. Наприклад, система машинного навчання може бути натренована на електронних повідомленнях для розрізнення спаму і прийнятних повідомлень. Після навчання вона може бути використана для класифікації нових повідомлень *електронної пошти* на спам та неспам. В основі машинного навчання розглядаються уявлення та узагальнення.

Оброблення природної мови – загальний напрям штучного інтелекту та лінгвістики. Він вивчає проблеми комп'ютерного аналізу та синтезу природної мови. Стосовно штучного інтелекту аналіз означає розуміння мови, а синтез – генерацію розумного тексту. Розв'язання цих проблем буде означати створення зручнішої форми взаємодії комп'ютера та людини.

У штучному інтелекті взаємозв'язок між задавалося б різними напрямками науки виражено дуже сильно, і це пов'язано з філософською суперечкою про сильний і слабкий ШІ.

Нині у створенні штучного інтелекту спостерігається інтенсивний перелом усіх предметних галузей, які мають хоч якесь відношення до ШІ в базі знань. Практично всі підходи були випробувані, але до появи штучного розуму жодна дослідницька група так і не дійшла.

Дослідження ШІ влились у загальний потік технологій сингулярності (видового стрибка, експотенціального розвитку людини), таких як нанотехнологія, молекулярна біоелектроніка, теоретична біологія, квантова теорія, ноотропіки, екстремофіли тощо.

Кінцевою метою досліджень з питань «штучного інтелекту» є розкриття таємниць мислення та створення моделі мозку. Принципова можливість моделювання інтелектуальних процесів впливає з основного гносеологічного результату кібернетики, який полягає у тому, що будь-яку функцію мозку, будь-яку розумову діяльність, описану мовою з точно визначеною семантикою за допомогою скінченного числа слів, в принципі можна передати електронній цифровій обчислювальній машині (ЕЦОМ). Сучасні ж наукові уявлення про природу мозку дають підстави вважати, що принаймні в суто інформаційному

аспекті найістотніші закономірності мозку визначаються скінченною (хоч, може, й надзвичайно великою) системою правил.

2.4 Технології створення інформаційних систем з елементами штучного інтелекту

Під інтелектуальними технологіями розуміють сукупність математичних методів і засобів проектування інформаційних систем (ІС) на основі CASE–інструментів. CASE–технологія (*computer-aided software engineering*) – набір інструментів і методів програмної інженерії, який забезпечує високу якість програмних продуктів.

Технологія, основана на знаннях, або інтелектуальна технологія, передбачає впровадження в ІС елементів штучного інтелекту, зокрема, баз знань і правил виведення для оброблення якісної інформації та природної мови. До інструментальних засобів інтелектуальних технологій відноситься продукт INTELLECT від «AI Corp», який надає змогу збирати, надавати й аналізувати дані відповідно до запитань англійською мовою.

Застосування інтелектуальних систем управління припускає у разі розроблення такої системи приділяти значну увагу засобам підвищення надійності, використання автоматизованих процедур і алгоритмів. У рамках цієї схеми широко використовуються прийоми агрегування та методи декомпозиції.

Опис законів функціонування інтелектуальної системи задається трьома сімействами функцій:

- функціями, що визначають зміни станів елементів системи $S(t)$;
- функціями, що задають вихідні сигнали елементів $U(t)$;
- функціями, що викликають зміни в структурі інтелектуальної системи.

Для формування повного опису інтелектуальної системи необхідно крім функцій $S(t)$, $U(t)$, $W(t)$ задати початковий стан системи, тобто початкову структуру W_0 і початкові стани всіх її елементів S_0 . Для дослідження дискретних систем основним інструментом є апарат теорії алгоритмів, автоматів, інформації.

Складність опису ІС визначається двома основними факторами: розмірністю (числом елементів і параметрів, що їх описують) і складністю структури, що обумовлена загальним числом зв'язків між її елементами і їхньою розмаїтістю. Складні системи управління – це системи з описами, що не зводяться до опису одного типового елемента і вказівки загального числа таких елементів. У разі визначення складності таких систем використовують різні моделі і методи моделювання, у тому числі декомпозицію та агрегування. Однак у цьому напрямку можливості моделювання обмежені. Відома теза Джона фон Неймана свідчить про те, що існує поріг складності системи, після досягнення якого найпростішим описом моделі стає сама система.

Схема функціонування інтелектуальної системи, у тому числі довільної системи управління, у найбільш загальному вигляді зображується як кругообіг інформації з такою динамікою, що забезпечує нормальне функціонування об'єкта. У цьому разі система управління видає керуючі впливи на об'єкт по каналу прямого зв'язку, результати такого впливу відображаються об'єктом управління, фіксуються та передаються в систему управління по каналу зворотного зв'язку, потім формується новий керуючий вплив, і цикл керування повторюється знову. Використання зворотного зв'язку, відоме під назвою «принцип зворотного зв'язку», – це фундаментальний принцип побудови інтелектуальної системи підтримки управління.

Все розмаїття інтелектуальних систем, які мають прикладне значення, можна класифікувати за різними ознаками, основною з яких є сфера застосування. Число таких сфер надзвичайно велике, відзначимо найважливіші: соціальні, економічні, технічні й біологічні системи. Всі вони належать до класів інформаційних систем, і в них реалізуються функції управління. Управління – це зміна стану системи, що веде до досягнення поставленої мети. Цілями управління можуть бути: підтримка деякого бажаного стану (рівня) системи у разі дії різного роду впливів, що їх збурюють; забезпечення режиму роботи енергосистеми, при якому досягається максимум виробництва електроенергії, або мінімум собівартості 1 кВт·год. Управління також можна визначити як вибір однієї із множини можливих альтернатив забезпечення нормального функціонування системи.

Для системи управління необхідно знати і передбачати її поведінку у разі можливих різних впливів на неї, для цього необхідно мати у своєму розпорядженні модель системи. Управляючі системи прикладного плану досить складні і для забезпечення свого функціонування вимагають дотримання певних принципів.

Логіка функціонування інтелектуальної системи визначає певні принципи побудови та забезпечення функціонування ІС. Серед основних принципів можна виділити наступні:

1. Визначення мети управління у разі забезпечення стійкості системи.
2. Обов'язковий облік можливих несприятливих зовнішніх впливів, які можуть вивести інтелектуальну систему зі стійкої рівноваги.
3. Необхідність методичного забезпечення засобів і алгоритмів управління з використанням:
 - математичного апарата штучного інтелекту;
 - систем підтримки прийняття управлінських рішень;
 - систем оцінки наслідків прийняття пропонованих рішень;
 - сполучення макро – і мікромоделей;
 - алгоритмів навчання, самонавчання та самовдосконалення;
 - оптимізаційних алгоритмів.

4. Розробка засобів контролю виконання програми управління на основі використання каналів зворотного зв'язку, систем моніторингу і контролю.

Для реалізації цих принципів одним з доцільних є використання програмної оболонки Decision Grid (розробник – «Softkit Technologies Inc»), яка має багато функціональних можливостей і створює зручні умови для кращого оцінювання та порівняння альтернатив. До основних переваг Decision Grid можна віднести можливість автоматичного розрахунку ваги критеріїв, визначення значень добору, виконання аналізу чутливості, перегляд множинних сценаріїв, графічний перегляд результатів за допомогою програмного інтерфейсу OLE Automation. У цьому разі даного програмного середовища для здійснення інтелектуальної підтримки управління ІС основними факторами успіху є:

- зручність використання, зв'язність даних,
- технологічна сумісність, легкий супровід і додаткова вартість.

Інтелектуальні системи або системи штучного інтелекту ШІ базуються на методах нелінійної динаміки, яка включає принципи роботи нейронних мереж НМ, нечіткої логіки НЛ, генетичних алгоритмів ГА, що моделюють функції людини (розумові, нервові та генетичні) на основі використання обчислювальних алгоритмів.

Нейронні мережі у загальному випадку – це певні методи (алгоритми) імітації процесів і явищ, що моделюють нервову систему людини і дають змогу відтворювати та обробляти складні залежності.

Нейронні мережі, виділені з ШІ в окремий напрям, набули широкого поширення завдяки здатності адаптуватися до різного роду ситуаційних завдань контролю і управління. Вони є основою нейромережових технологій.

Під *нейромережовими* технологіями розуміють апаратно і програмно реалізовані блоки обробки інформації на базі нейронних мереж, під *обчислювальними технологіями* – реалізацію оптимальних методів і алгоритмів роботи мікропроцесорними засобами. Нейромережові технології тісно переплітаються з обчислювальними технологіями, які відносяться до галузі інформатики, що займається «м'якими» обчисленнями, тобто мають справу з даними, які характеризує неповнота і відсутність точності.

Обчислювальні технології базуються на принципах нечіткої логіки і НМ, еволюційних та генетичних алгоритмів, а також на деяких розділах теорії вірогідності.

Нечіткі множини – це множини з різним ступенем належності до неї її елементів (від 0 до 1). У класичних множинах кожний її елемент має належність до неї на рівні 1.

Нечітка логіка базується на нечітких множинах і розповсюджує достовірність (істинність) дії в межах від 0 до 1, тобто дія може не тільки відбутися або не відбутися (відповідно до класичної логіки Буля), а і відбутися з певним ступенем достовірності в межах від 0 до 1. У загальному випадку – це

певні методи (алгоритми) імітації процесів і явищ, що моделюють розумову функцію людини і дають змогу відтворювати та обробляти складні залежності у вигляді лінгвістичних залежностей.

Еволюційні алгоритми – напрям в ШІ (розділ еволюційного моделювання), який використовує імітацію біологічної еволюції і моделює процеси відбору, мутації і відтворення корисних ознак системи методами машинних алгоритмів.

Генетичний алгоритм – це евристичний алгоритм пошуку, який використовується для вирішення завдань оптимізації і моделювання шляхом випадкового підбору, комбінування і варіації шуканих параметрів системи з використанням механізмів, що нагадують біологічну еволюцію. Відмінна особливість генетичного алгоритму від еволюційного – акцент на використання оператора «схрещування», що формує операцію рекомбінації рішень–кандидатів, роль якої аналогічна ролі схрещування в живій природі.

Евристика (від давньогрецького *εὐρίσκω* – «відшукую», «відкриваю») – сукупність логічних прийомів, методів і правил, що полегшують і спрощують рішення пізнавальних, конструктивних, практичних завдань. Евристика – це момент відкриття нового, а також методи, які використовуються в процесі цього відкриття.

Для забезпечення системного аналізу у разі створення інтелектуалізованих ІС доцільно використовувати такі сучасні математичні методи щодо моніторингу та накопичення даних і перетворення їх на знання: дейтамайнінг, кластеризація, нечітка логіка, нейронні мережі, генетичні алгоритми та програмні агенти. За своїми принципами роботи ці методи відносяться до нелінійних.

Дейтамайнінг, або процес фільтрування великих обсягів даних використовується для того, щоб підбирати відповідну для контексту задачі інформацію, є великою цінністю для керівників і аналітиків у їх повсякденній діяльності. Сучасними інструментальними засобами дейтамайнінгу сьогодні є програмні продукти PolyAnalyst, MineSet, KnowledgeSTUDIO.

Доцільним є використання нейронних мереж – програмно реалізовані системи, в основу яких покладені математичні моделі процесу передавання і оброблення імпульсів мозку людини, що імітують механізми взаємодії нейронів з метою опрацювання інформації, яка надходить, і навчання для набуття досвіду. Найбільш популярним програмним пакетом, який реалізовує нейромережевий підхід, є NeuroShell.

Програмним пакетом, який реалізовує математичний апарат нечіткої логіки, є додаток Fuzzy програмного середовища MatLab та програмний пакет FuzzyTech.

Генетичні алгоритми, які можна вважати одним із видів дейтамайнінгу, представлені такими популярними пакетами, як Evolver (знаходження оптимального рішення в таблицях Excel – <http://www.palisade.com/evolver/ru/>),

GeneHunter (знаходить популяцію можливих рішень проблеми оптимізації з таблиць Microsoft Excel за допомогою GA <http://www.wardsystems.com/genehunter.asp>), Genetic Training Option (оптимізації синапсів нейронних мереж <https://calsci.com/GTO.html>), які сприяють розширенню галузей застосування інтелектуальних систем і для ефективного застосування потребують від користувачів тільки початкової формалізації задачі та формування множини початкових даних.

КОНТРОЛЬНІ ЗАПИТАННЯ ТА ЗАВДАННЯ

1. *Методи створення систем штучного інтелекту.*
2. *Наведіть визначення штучного інтелекту.*
3. *Підходи до розуміння задач штучного інтелекту.*
4. *На чому базується штучний інтелект?*
5. *Напрями досліджень штучного інтелекту.*
6. *Наведіть визначення нечіткої множини.*
7. *Наведіть визначення нейронної мережі.*
8. *Наведіть визначення генетичного алгоритму.*

ЛІТЕРАТУРА

[14], стор. 5–10; [28], стор. 15–17; [7], стор. 5–9; [3], стор. 11–26; [4], стор. 10–25; [34], стор. 3,4; [14], стор. 5–10; [19], стор. 6–20; [16], стор. 7–9.

РОЗДІЛ 3

КЛАСИЧНІ МЕТОДИ УПРАВЛІННЯ В ЕЛЕКТРОЕНЕРГЕТИЧНИХ СИСТЕМАХ

3.1 Системи прийняття рішень щодо управління на основі класичних методів

Для представлення систем управління використовуються структурні схеми.

Структурною схемою в теорії автоматичного управління називають графічне зображення математичної моделі системи управління у вигляді з'єднань ланок.

Ланку на структурній схемі умовно позначають у вигляді прямокутника з позначкою вхідних і вихідних величин, а також передатної функції всередині нього. Іноді замість передатної функції вказують рівняння або характеристику. Ланки можуть бути пронумеровані, а їхні передатні функції, рівняння або характеристики представлені поза структурною схемою.

Вхідні й вихідні величини записують у вигляді зображень, якщо передатні функції задають у формі зображень. Якщо ж передатні функції задають в операторній формі або ланки описують диференціальними рівняннями, то вхідні і вихідні змінні записують у вигляді оригіналу.

Порівняльні (рис. 3.1, *а, б*) і підсумовуючі (рис. 3.1, *в*) ланки зображують у вигляді кола, розділеного на сектори. У ланці сектор, на який подається від'ємний сигнал, затемнюють (рис. 3.1, *б*) або перед відповідним входом ставлять знак мінус (рис. 3.1, *а*).

Структурні схеми широко використовують на практиці у разі дослідження й проектування автоматичних систем керування, оскільки вона дає наочне представлення про зв'язки між ланками, проходження і перетворення сигналів у системі.

У разі математичного опису автоматичну систему звичайно зображують у вигляді блок–схеми й для кожного «блока» (елемента) записують рівняння, виходячи з фізичних законів, яким підкоряються процеси в ньому. Структурну схему можна скласти на основі цієї блок–схеми й отриманих рівнянь або тільки на основі останніх. Подальші перетворення, необхідні для одержання рівнянь і передатних функцій системи, простіше й наочніше робити за структурною схемою.

Ланка на структурній схемі не обов'язково зображує модель якого–небудь окремого елемента. Вона може бути моделлю елемента, з'єднання елементів або взагалі будь–якої частини системи.

Розглянемо основні правила перетворення структурних схем.

1. *Послідовне з'єднання ланок* (рис. 3.2, а). За послідовного з'єднання вихідна величина кожної передатної ланки є входнім впливом наступної ланки. У випадку перетворення структурних схем ланцюг з послідовно з'єднаних ланок можна замінити однією ланкою (рис. 3.2, б) з передатною функцією $W(s)$, яка дорівнює добутку передатних функцій окремих ланок:

$$W(s) = \prod_{i=1}^n W_i(s).$$

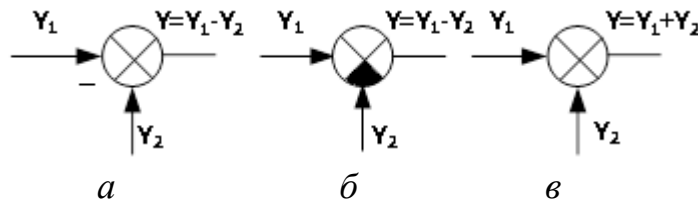


Рис. 3. 1. Порівняльні (а, б) і підсумовуючі (в) ланки

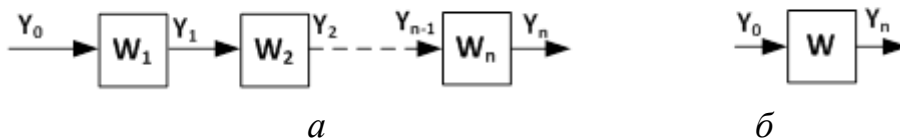


Рис. 3. 2. Послідовне з'єднання ланок

Запишемо рівняння ланок: $y_1 = W_1 y_0$; $y_2 = W_2 y_1$; $y_n = W_n y_{n-1}$.

Виключивши з цієї системи змінні, одержимо $y_n = W_1 \cdot W_2 \dots W_n y_0$.

Звідки $W = \prod_{i=1}^n W_i$.

2. *Паралельне з'єднання ланок* (рис. 3.3, а). За паралельного з'єднання на вхід всіх ланок подається той самий сигнал, а вихідні величини складаються. Ланцюг з паралельно з'єднаних ланок можна замінити однією ланкою (рис. 3.3, б) з передатною функцією $W(s)$, яка дорівнює сумі передатних функцій входніх у неї ланок: $W(s) = \sum_{i=1}^n W_i(s)$.

Для виведення цієї формули складемо рівняння для кожної ланки: $y_1 = W_1 y_0$; $y_2 = W_2 y_0$; $y_n = W_n y_0$.

Склавши ці рівняння і з огляду на те, що $y = \sum_{i=1}^n y_i$, одержимо шукану формулу.

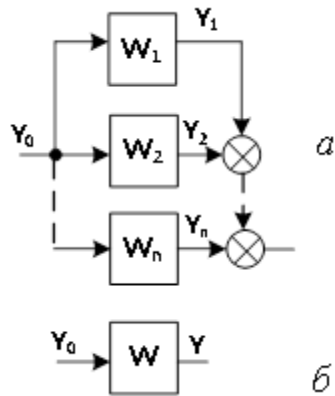


Рис. 3.3. Паралельне з'єднання ланок

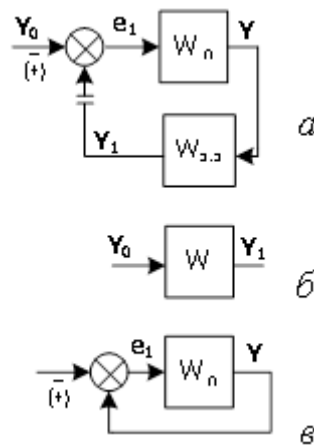


Рис. 3.4. Ланка, охоплена зворотним зв'язком

3. *Ланка, охоплена зворотним зв'язком* (рис. 3.4, а). Прийнято вважати, ланка охоплена зворотним зв'язком, якщо її вихідний сигнал через яку-небудь іншу ланку подається на її вхід. У цьому разі, якщо сигнал y_1 зворотного зв'язку віднімається із вхідного впливу y_0 , тобто $e_1 = y_0 - y_1$, то зворотний зв'язок називають *негативним*. Якщо сигнал y_1 зворотного зв'язку додається до вхідного впливу y_0 , тобто $e_1 = y_0 + y_1$, то зворотний зв'язок називають *позитивним*.

Розіркнемо зворотний зв'язок перед ланкою, що порівнює сигнали (рис. 3.4, а). Тоді одержимо ланцюг із двох послідовно з'єднаних ланок. Тому передатна функція W розіркнутого ланцюга (рис. 4,а) дорівнює добутку передатної функції W_{Π} прямого ланцюга і передатної функції $W_{3.3}$ зворотного зв'язку: $W = W_{\Pi} \cdot W_{3.3}$ (рис. 3.4, б).

Передатна функція W_3 замкнутого ланцюга з негативним зворотним зв'язком (ланки, охопленої негативним зворотним зв'язком), дорівнює передатній функції прямого ланцюга, діленої на одиницю, плюс передатна функція розіркнутого ланцюга:

$$W_3 = \frac{W_{\Pi}}{(1 + W)}$$

Для висновку цієї формули випишемо рівняння для кожної ланки:

$$y = W_{\Pi} \cdot e_1; \quad y_1 = W_{3.3} y; \quad e_1 = y_0 - y_1.$$

У цій системі останнє рівняння для e_1 називають *рівнянням замикання*.

Замінивши змінні e_1 і y_1 в наведеній системі, одержимо рівняння

$$y = W_{\Pi} \cdot (y_0 - W_{3.3} y)$$

$$\text{або} \quad (1 + W_{\Pi} \cdot W_{3.3}) y = W_{\Pi} y_0.$$

Звідси

$$W_3 = \frac{W_{\Pi}}{(1 + W_{\Pi} \cdot W_{3.3})} = \frac{W_{\Pi}}{(1 + W)}.$$

Якщо зворотний зв'язок позитивний, то аналогічно одержимо

$$W_3 = \frac{W_{\Pi}}{(1 - W)}$$

Таким чином, *передатна функція замкнутого ланцюга з позитивним зворотним зв'язком дорівнює передатній функції прямого ланцюга, поділеній на одиницю, мінус передатна функція розімкнутого ланцюга.*

Розглянемо обчислення передатної функції одноконтурної системи.

Замкнуту систему (структурну схему) називають *одноконтурною*, якщо у разі її розмикання в якій-небудь точці виходить ланцюг з послідовно з'єднаних ланок або ланцюгів, що не містять паралельних і зворотних зв'язків.

Розглянемо одноконтурну систему, наведену на рис. 3.5, а. Знайдемо передатну функцію за входом g і виходом y . Ділянку за ходом сигналу від точки вхідного впливу до точки знімання вихідного сигналу назвемо *прямим ланцюгом* (рис. 3.5, а), а ланцюг з послідовно з'єднаних ланок, що входять у замкнутий контур (рис. 3.5, б), – *розімкнутим ланцюгом*. Для перевірки справедливе таке правило: *передатна функція одноконтурної системи з негативним (позитивним) зворотним зв'язком дорівнює передатній функції прямого ланцюга, діленій на одиницю, плюс (мінус) передатна функція розімкнутого ланцюга:*

$$W_{yg} = \frac{W_0 \cdot W_1 \cdot W_2}{1 \pm W_1 \cdot W_2 \cdot W_3} = \frac{W_{\Pi}}{1 \pm W},$$

де W_{Π} – передатна функція прямого ланцюга;

W – передатна функція розімкнутого ланцюга.

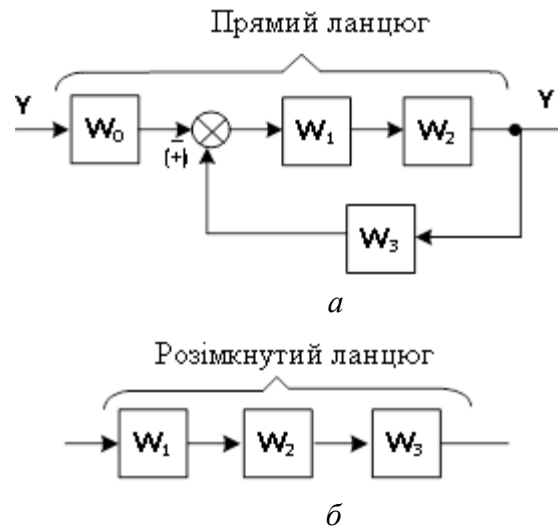


Рис. 3.5. Ланцюги: а – Прямий; б – розімкнутий

Сформульоване правило справедливе для будь-якої одноконтурної системи.

Обчислення передатної функції багатоконтурної системи. Замкнену систему (структурну схему) називають *багатоконтурною*, якщо при її розмиканні виходить ланцюг, що містить паралельні або зворотні зв'язки, або інакше замкнену систему називають *багатоконтурною*, якщо вона крім головного зворотного зв'язку містить місцеві зворотні або паралельні зв'язки. Вважають, що багатоконтурна система має *перехресні* зв'язки, якщо контур зворотного або паралельного зв'язку охоплює ділянка ланцюга, що містить тільки початок або кінець іншого ланцюга зворотного або паралельного зв'язку (рис. 3.6, а, б).

Для обчислення передатної функції багатоконтурної системи необхідно насамперед перестановкою й перенесенням вузлів і суматорів звільнитися від перехресних зв'язків.

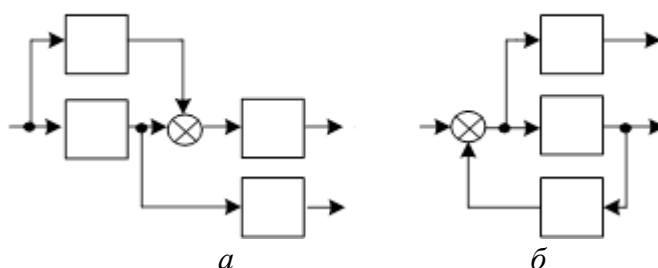


Рис. 3.6. Багатоконтурна система

Потім, використовуючи перші три правила перетворення структурних схем, перетворити її в одноконтурну систему, передатну функцію якої легко обчислити відповідно до сформульованого вище правила. Треба мати на увазі, що при перетворенні структурної схеми не можна переносити суматор через точку знімання вихідного сигналу, оскільки у цьому випадку точка знімання виявляється на нееквівалентній ділянці лінії зв'язку.

На рис. 3.7 і 3.8 представлена архітектура компонентів процесу класичного та інтелектуального управління. Як класичний, так і інтелектуальний контролери формують зворотний зв'язком в систем управління.

Система управління зі зворотним зв'язком – це така система, що прагне підтримувати встановлену залежність між двома змінними шляхом порівняння функцій від цих змінних і використання різниці між ними, для того щоб ініціювати керуючий вплив.

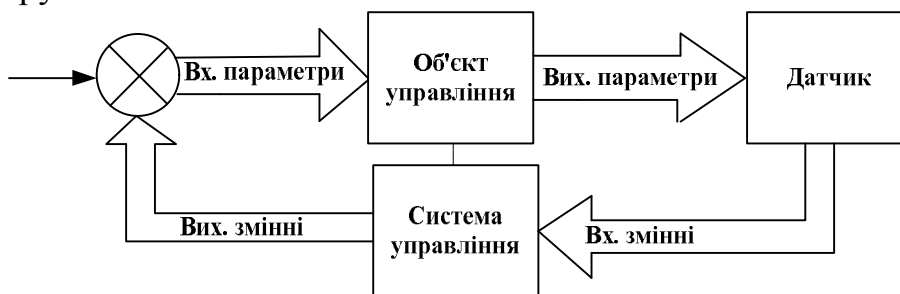


Рис. 3.7. Архітектура компонентів процесу класичного управління

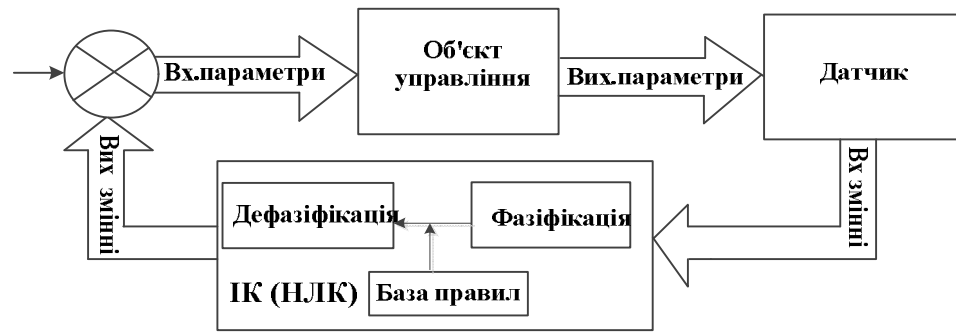


Рис. 3.8. Архітектура компонентів процесу управління з інтелектуальним контролером (нечітким логічним контролером)

Розглянемо *класичні (традиційні) методи управління*, які реалізуються регуляторами в електроенергетичних системах. *Регулятор* – це пристрій, призначений для підтримки контрольованого параметра на рівні заданого значення в складі автоматизованої системи управління.

Регулятор виробляє на виході сигнал Y , який спрямований на зменшення відхилення E , тобто різниці між контрольованою величиною і заданим значенням. Цей сигнал плавно змінюється від 0 до 100 % і може подаватися у вигляді струму або імпульсів на виконавчий пристрій (нагрівач, електродвигун, холодильник і ін.). Якщо в якості вихідних пристроїв приладів використовуються реле, перехід від аналогового вихідного сигналу Y до релейного (імпульсного) відбувається за допомогою широтно-імпульсної модуляції (ШІМ), рис. 3.9. Величина вихідного сигналу Y перетвориться в тривалість релейних імпульсів D стосовно періоду їх повторення, тобто

$$Y = \frac{D}{T_{\Pi}} \cdot 100 \% , \quad D = Y \cdot \frac{T_{\Pi}}{100 \%} ,$$

де D – тривалість імпульсу;
 T_{Π} – період проходження імпульсів.

Розглянемо *пропорційний метод управління (П-регулятор)*.

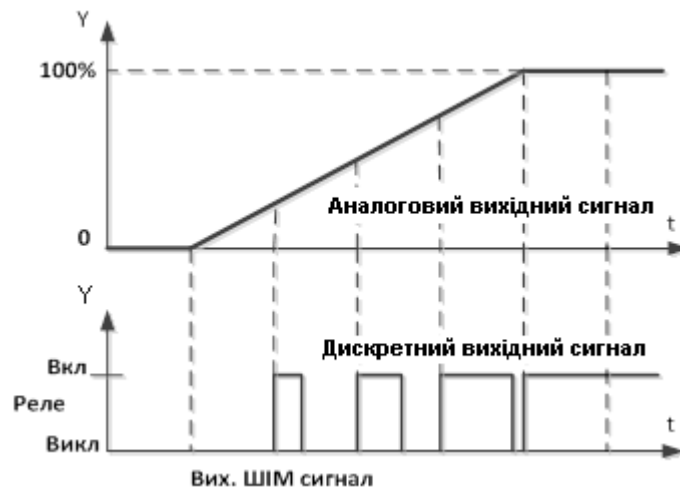
У разі пропорційного методу управління вихідний сигнал регулятора Y пропорційний величині відхилення E , тобто

$$Y = \frac{1}{X_{\Pi p}} \cdot E \cdot 100 \% ,$$

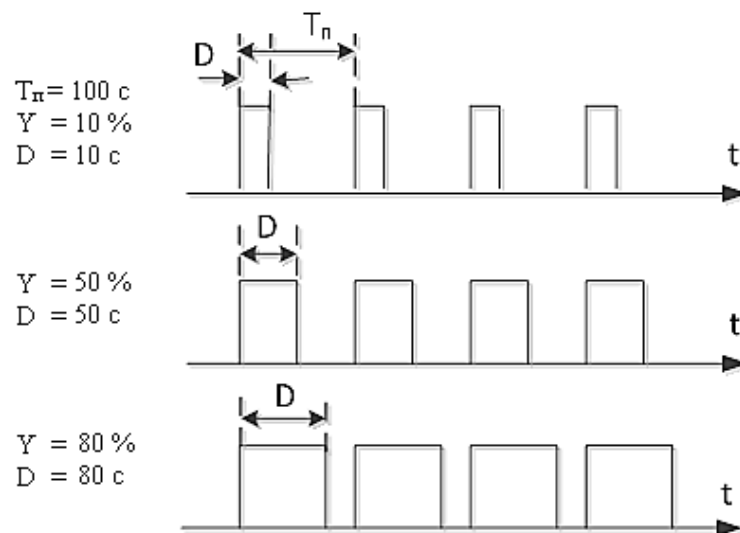
де $X_{\Pi p}$ – смуга пропорційності, у межах якої ця формула відповідає.

Смуга пропорційності $X_{\Pi p}$, як і відхилення E , виражається в одиницях контрольованого параметра і визначає величину вихідного сигналу Y . Чим

ширше смуга пропорційності $X_{пр}$, тим менше величина вихідного сигналу Y за того ж відхилення E .



a



б

Рис. 3.9. Вихідні сигнали: *a* – аналоговий та дискретний; *б* – дискретний з широтно–імпульсною модуляцією для різних значень тривалості імпульсу

Поза смугою пропорційності вихідний сигнал Y дорівнює 0 або 100 %.

У випадку використання *пропорційного методу* регулятор формує імпульси, у яких присутня тільки пропорційна складова величини вихідного сигналу. На рис. 3.10 представлена графічна інтерпретація формування вихідного сигналу пропорційним методом з використанням широтно–імпульсної модуляції та аналогового сигналу.

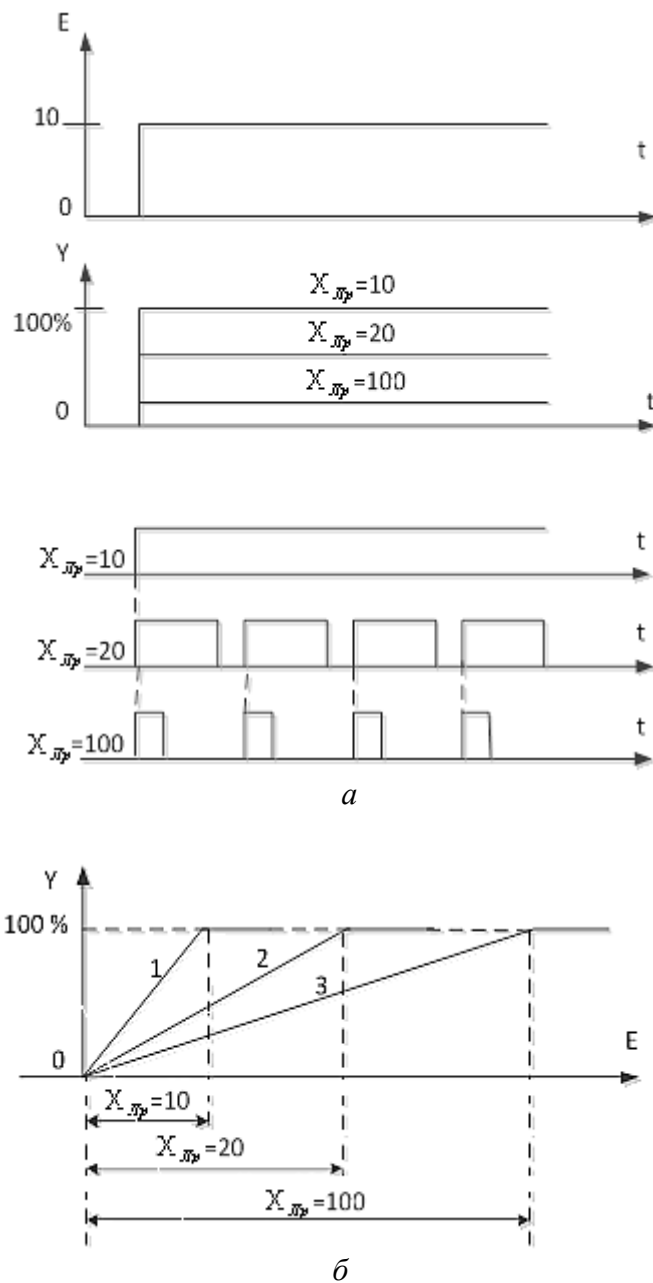


Рис. 3.10. Пропорційне регулювання: *а* – дискретний вихід;
б – аналоговий вихід

Розглянемо *пропорційно-диференціальний* метод управління (*ПД-регулятор*).

У випадку роботи регулятора в цьому режимі величина вихідного сигналу Y залежить не тільки від величини відхилення E , але й від швидкості її зміни. Оскільки регульована величина вимірюється не постійно, а періодично, формула вихідного сигналу має такий вигляд:

$$Y = \frac{1}{X_{Pr}} \cdot \left(E_i + t_o \frac{\Delta E}{\Delta t_{вим}} \right) \cdot 100 \% ,$$

де $X_{пр}$ – смуга пропорційності; E – відхилення; t_d – постійна часу диференціювання; ΔE – різниця між двома сусідніми вимірами E ; $\Delta t_{вим}$ – час між двома сусідніми вимірами; $\Delta E / \Delta t$ – швидкість зміни відхилення E .

Зміна вихідного сигналу регулятора Y за ступінчатої зміни відхилення E показана на рис. 3.11, а. За перший період після ступінчатої зміни відхилення E регулятор видає керуючий імпульс, у якому присутня пропорційна складова, викликана зміною E (не заштрихована частина першого імпульсу). Величина диференціальної складової залежить від величини ΔE і коефіцієнта t_d . У наступних імпульсах присутня тільки пропорційна складова, оскільки немає зміни E .

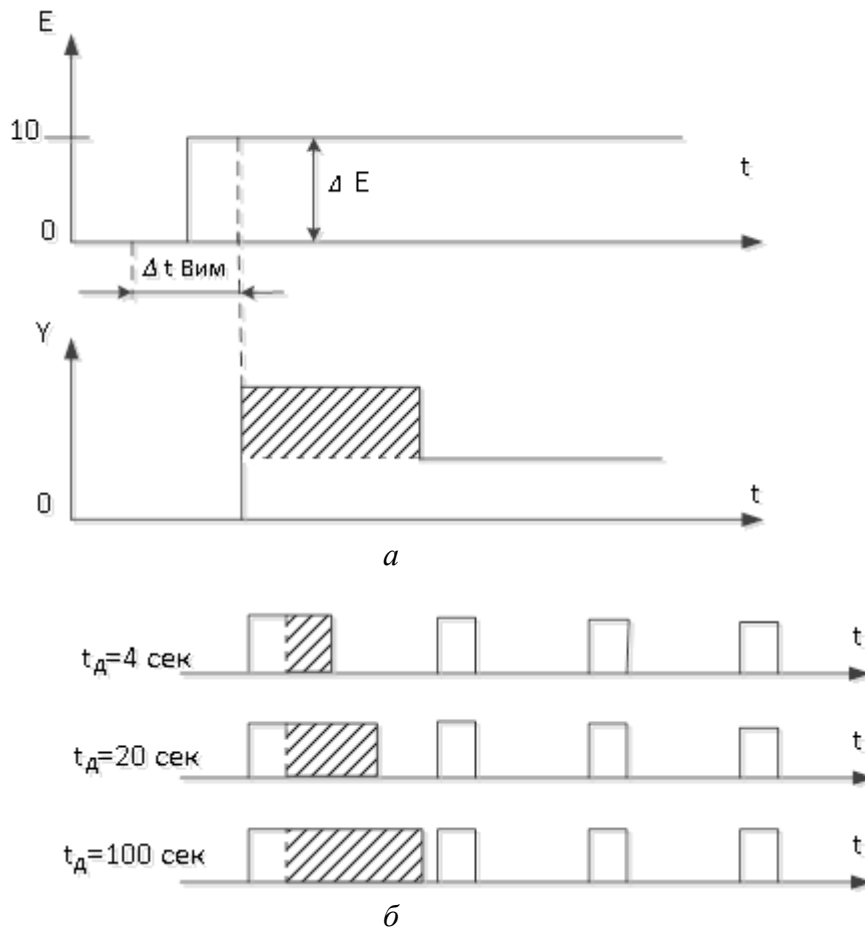


Рис. 3.11. Пропорційно–диференціальне регулювання: *а* – зі зміною вихідного сигналу за амплітудою; *б* – зі зміною вихідного сигналу за скважністю

Розглянемо *пропорційно–інтегральний* метод управління (*ПІ–регулятор*).

У разі роботи регулятора в цьому режимі величина вихідного сигналу Y залежить як від величини відхилення E , так і від часу його існування.

Оскільки регульована величина вимірюється не неперервно, а періодично, формула вихідного сигналу має такий вигляд:

$$Y = \frac{1}{X_{\text{пр}}} \cdot \left(E_i + \frac{1}{\tau_{\text{инт}}} \sum_{n=0}^{n=i} E_n \right) \cdot 100 \%,$$

де $X_{\text{пр}}$ – смуга пропорційності; E_i – відхилення; $\tau_{\text{инт}}$ – стала часу інтегрування; $\sum_{n=0}^{n=i} E_n$ – сумарна різниця між сусідніми вимірами відхилення E . На рис. 3.12 представлено графічну інтерпретацію формування вихідного сигналу пропорційно–інтегральним методом управління.

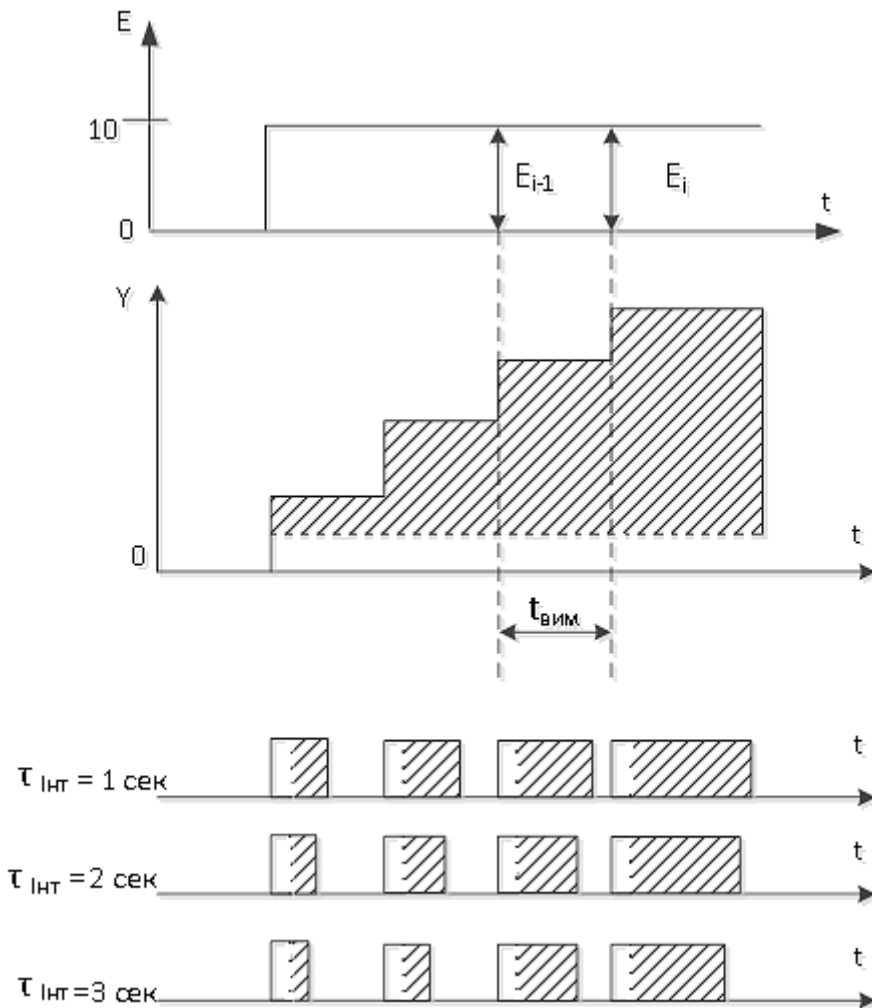


Рис. 3.12. Пропорційно–інтегральне регулювання

З рисунка видно, що спочатку, коли немає відхилення ($E=0$), немає й вихідного сигналу ($Y=0$). З появою відхилення E_i з'являються імпульси, тривалість яких поступово збільшується. В імпульсах існує пропорційна складова, яка залежить від величини E (незаштрихована частина імпульсів) та інтегральна складова (заштрихована частина). Збільшення тривалості імпульсів відбувається за рахунок росту інтегральної складової, яка залежить від відхилення E і коефіцієнта $\tau_{\text{инт}}$.

Розглянемо *пропорційно–інтегрально–диференціальний* метод управління (ПІД–регулятор).

Для реалізації цього методу вихідний сигнал регулятора Y змінюється відповідно до виразу

$$Y = \frac{1}{X_{Pr}} \cdot \left(E_i + t_d \frac{\Delta E}{\Delta t_{вим}} + \frac{1}{\tau_{int}} \sum_{n=0}^{n=i} E_n \right) \cdot 100\%,$$

де всі складові формули описані вище.

У вихідних сигналах управління присутні всі три складові – пропорційна, диференціальна, інтегральна, які змінюються за законами, описаними вище.

ПІД–регулювання є найбільш ефективним із усіх описаних.

Класичні методи управління відносяться до лінійних, які не в змозі забезпечити прийняття рішень щодо управління в широкому діапазоні режимних параметрів ЕЕС, для яких характерним є їх взаємопов'язаність, нелінійність та непрогнозованість.

КОНТРОЛЬНІ ЗАПИТАННЯ ТА ЗАВДАННЯ

1. Наведіть визначення передатної функції.
2. Що таке структурна схема?
3. Дайте визначення ланки зворотного зв'язку.
4. Який буває зворотний зв'язок?
5. Що таке статичний та астатичний регулятор?
6. Дайте визначення регулятора та його передатної функції.
7. Що таке структурна схема?
8. Особливості пропорційного методу регулювання.
9. Особливості пропорційно–диференціального методу регулювання.
10. Характеристика інтегрального методу регулювання.
11. Передатна функція пропорційно–інтегрально–диференціального регулятора.

ЛІТЕРАТУРА

[9], стор. 5–14; [18], стор. 13–33, 64–74

РОЗДІЛ 4

МАТЕМАТИЧНИЙ АПАРАТ НЕЧІТКОЇ ЛОГІКИ ТА ЙОГО ВИКОРИСТАННЯ В ЕЛЕКТРОЕНЕРГЕТИЧНИХ ЗАДАЧАХ

4.1 Аналіз і характеристика підходів до створення систем з елементами штучного інтелекту

Структура більшості систем регулювання та стабілізації параметрів (у тому числі й з використанням мікропроцесорної техніки), що на сьогодні має місце в електроенергетичній системі, подібна до традиційних локальних систем з ПД–методом управління і не може забезпечити оптимальне управління та прийняття адекватних рішень без математичного опису системи. Відомі системи з прогнозним (упередженим) керуванням та так звані «робастні» системи теж вимагають наявності детермінованого причинно–наслідкового зв'язку між вхідними і вихідними параметрами у вигляді математичних рівнянь, які для більшості промислових електротехнологічних процесів неможливо визначити. За відсутності достовірного математичного опису поведінки системи в умовах непрогнозованої зміни параметрів мережі неможливо оптимально встановити коефіцієнти управління класичного ПД–регулятора, який має працювати на нелінійне навантаження.

До нелінійних об'єктів належать і електричні мережі та більшість промислових електротехнологічних установок, у тому числі електронасоси, електропечі опору, компенсуючі пристрої тощо, які широко використовуються в енергетиці. Для здійснення технологічного циклу в нелінійних навантаженнях необхідно протягом багатьох годин виконувати регулювання напруги у всьому вхідному діапазоні напруги мережі живлення. Тому у разі реалізації необхідної якості технологічного процесу в установках з розосередженими параметрами виникає проблема визначення та багаторазового коригування коефіцієнтів управління протягом технологічного циклу.

Нині одним із ефективних шляхів у вирішенні проблеми регулювання параметрів електроенергетичних режимів є розробка систем автоматизованого управління з використанням в них інтелектуальних методів на основі математичного апарата нечіткої логіки, які реалізують управління на основі змодельованих міркувань експерта і не потребують математичного опису поведінки системи.

Три десятиліття тому в системах управління крім «жорсткої» логіки стали використовувати мікропроцесори, а потім і мікроконтролери. Швидке вторгнення ЕОМ в життя людини, їх мініатюризація та широка доступність призвели до зміщення акценту в бік програмних засобів у разі створення та реалізації систем управління. Це дало змогу максимально гнучко використовувати набуті енергетичні та часові характеристики, багаторазово

збільшувати ефективність традиційних електротехнологічних систем і значно перекидати витрати на впровадження автоматизації. Тенденція останнього часу (як програмна, так і апаратна) прагне до максимальної уніфікації (конструктивів, інтерфейсів, алгоритмів і т.п.), в ідеалі приводячи до інтегрованої конструкції. Створення зразків нової техніки та розумних електричних мереж з інтелектуальним управлінням на основі штучного інтелекту, зокрема, математичного апарата нечіткої логіки набуває характерних та актуальних рис, які за всіма системними оцінками будуть розвиватися.

Інтелектуальні нечіткі системи прийняття рішень викликали нові напрямки стратегій управління та розширення не тільки класичних традиційних методів управління, а і розширення підходів до створення топології побудови електроенергетичних систем, які здатні до «примітивного мислення», на основі накопичення знань та формування оптимальних з погляду людини рішень управління.

З точки зору інформаційних технологій нечітке управління базується на правилах, сформованих експертом, що має певний досвід у тому чи іншому технічному напрямку, а з точки зору технології систем управління нечітке управління – це регулювання з нелінійною передатною характеристикою.

Значне просування в напрямку розвитку теорії наближених міркувань було зроблено у 1963 році професором каліфорнійського університету (Берклі, США) Лотфі А. Заде (Lotfi A.Zadeh). Його робота «Fuzzy sets» у часопису «Information and control» заклала основи математичного апарата моделювання інтелектуальної діяльності людини і стала початковим поштовхом до розвитку принципово нової математичної теорії. Суть її полягає в розширенні класичного поняття множини за рахунок припущення того, що характеристична функція (функція належності елемента до множини) може набувати будь-яких значень в інтервалі від 0 до 1, а не тільки значень 0 та 1. Кожний елемент a_i такої множини супроводжується конкретним числовим значенням функції належності (істинності) μ_i та записується у вигляді пари (кортежу) a_i / μ_i . Такі множини одержали назву нечітких (fuzzy). Л.Заде визначив ряд операцій над *нечіткими множинами*.

Введення поняття лінгвістичної змінної та припущення того, що в якості її значень (термінів) виступають нечіткі множини, дало змогу створити математичний апарат для опису процесів інтелектуальної діяльності та відтворення моделі рішення людини, включаючи нечіткість та невизначеність виразів.

Винайдена в США нечітка логіка практичну розробку спочатку отримала в Європі, зокрема в Англії, у 1970 році для управління паровими котлами. В основному цією теорією, яку віднесли до спеціальної, займалися науково-дослідні організації; розробка її відбувалась на цьому етапі досить повільно. Але з 1980 року до теорії нечітких систем виявила інтерес Японія і дала могутній поштовх промисловим використанням нечітких систем управління,

далеко позаду залишивши Європу та Америку. Вже до 1990 року за цією тематикою було опубліковано близько 10 000 робіт. На сьогодні ціла низка часописів узагальнює розвиток теорії нечіткої логіки, зокрема, IEEE Transaction on Fuzzy Systems асоціації IEEE Computational Intelligence Society (CIS).

Зважаючи на те, що управління та обробка інформації в передових країнах орієнтуються і безперечно будуть у майбутньому орієнтуватися на комп'ютерні системи та обчислювальні алгоритми, можна стверджувати, що теорія нечіткої логіки та нечітких множин стане однією з основ створення розумних електроенергетичних систем Smart Grid.

Теорія нечіткої логіки – єдина теорія, яка математично оперує з думками, що вміщені в слова людини. Нечітка логіка дає змогу математично представити думки людини, тобто способи прийняття рішень управління та способи моделювання складних об'єктів, і, окрім цього, цілком придатна для представлення знань.

Людина в своїй повсякденній діяльності ніколи не користується формальним моделюванням на основі математичних виразів. Мова, якою користується людина для моделювання, – це і є нечітка природна мова. Людина не запам'ятовує своїх окремих дій, вона їх узагальнює та упорядковує у вигляді знань так, щоб їх легко було зрозуміти та використати в тій чи іншій ситуації; вона користується в своїх діях не числами, а словами «мало», «трохи більше», «лівіше» і таке ін. Це і є нечіткі слова управління діями людини. Саме такий підхід покладено в основу нечіткого управління. Всі входні чіткі параметри в нечіткому логічному контролері перетворюються в нечіткі слова–терми, а вже потім математично обробляються методами нечіткої логіки. Проте кінцевий результат управління все таки відпрацьовується у вигляді чіткого числового значення.

Тобто нечітке управління не слід розуміти дослівно і вважати, що воно розмите або просто ніяке. Нечітка множина строго визначається за допомогою функції належності. Іншими словами, логіка визначення поняття нечіткої множини не включає в себе будь-якої нечіткості. Нечітка логіка не взята невідомо звідки, вона опирається на функцію належності, яка чітко визначається за допомогою оціночних значень в інтервалі від 0 до 1.

Що ж дає така система управління на основі нечіткої логіки? Насамперед, якісний підхід до вирішення будь-якої задачі. Це не сліпе управління з використанням тих чи інших математичних рівнянь, а управління з поточним аналізом ситуації та змістовним прийняттям рішення в тій чи іншій ситуації, керуючись здоровим глуздом, досвідом та знаннями, тобто максимально адекватна реакція на всебічно оцінену похибку управління. Це мінімальні перехідні процеси. Це висока гнучкість, надійність і ефективність та спрощення управління в порівнянні з адаптивним та самонавчальними алгоритмами управління. Але головне це інтелектуальна система управління,

яка ставить за мету не просто управління заради самого управління, а забезпечення якості кінцевого продукту управління процесом.

Наприклад: у вузлах розподільних мереж електроенергетичних систем в умовах невизначеності за допомогою нечіткої логіки забезпечується формування пріоритетів першочергового переведення мереж напругою 6 кВ на напругу 20 кВ та оптимального використання інвестиційних планів; у системі електроприводів підйомних кранів нечітке керування забезпечується, без використання складних та дорогих у реалізації алгоритмів адаптивного управління зменшення розгойдування вантажу у разі його підйому та переміщення самого крана, усувається залежність роботи крана від маси, габаритів вантажу та умов підйому, за рахунок чого значно збільшується як швидкість підйому, так і швидкість переміщення крана, підвищується гарантія безпечності роботи; у стабілізаторах змінної напруги нечітке управління це не просто реалізація підтримки напруги на потрібному рівні, а підтримка її з можливою всебічною оцінкою та прогнозом ситуації в мережі з аналізом не тільки похибки та її знаку, а і першої та другої похідних, а також оцінкою величини і характеру навантаження споживача; у разі нечіткого управління електричним приводом у пральній машині головним є якість випраної білизни, з одночасним контролем за навантаженням барабана, його вібрацією, швидкістю та прискоренням його обертання, що призводить до зменшення споживання електроенергії та води.

Наведемо кілька наочних прикладів ефективного використання нечіткого управління у світовій практиці.

У місті Сендай (Японія) впровадження системи управління електропотягів метрополітену призвело до підвищення граничної швидкості руху потягів між станціями з 45 до 75 км/год. Також в Японії у разі використання нечіткої логіки за управління металургійними прокатними станами холодної прокатки точність прогнозування прокату підвищилася втричі за умови збільшення сили прокатки на 10 %.

З 1990 року Японія використовує нечітку логіку в системі управління кондиціонування повітря, мінімізуючи у цьому разі енергоспоживання за максимального комфорту, забезпечує економію електроенергії до 24 %. В світлу частину дня встановлюється одна температура, а в темну – друга. Враховується світить сонце чи ні, оскільки за однієї і тієї ж температури комфортніше, коли світить сонце.

У Німеччині заміна звичайних датчиків температури на датчики з нечіткою логікою в пристроях контролю температури обігрівання будинків дає щорічну економію 350 000 євро. А введення нечіткого регулювання в пральних машинах для оцінки прання супроводжується 20 %-вою економією води та електроенергії. У цьому випадку щорічна економія складає 400 000 євро.

Система управління на основі нечіткої логіки завжди зберігає ясність та простоту розуміння, тому що «інтелект» системи закладений не в

диференціальних рівняннях або вихідному тексті, а в змодельованих міркуваннях експерта на основі бази знань з визначеним набором правил поведінки (реакції) системи за тих або інших станів об'єкта управління.

4.2 Особливості організації структурних схем інтелектуальних систем з нечітким логічним контролером

Правильна організація структури та взаємодії всіх ланок інтелектуального контролера з максимально можливим урахуванням всіх вхідних параметрів впливу та аналізом властивостей і законів функціонування кожного компонента є запорукою формування правильного рішення управління силовим виконавчим органом. Тобто рішення щодо управління має прийматися, виходячи з існування цілісної структури системи, а взаємозв'язок має формуватися на етапі розробки концепції побудови системи і її складових на основі моделювання як нечіткого логічного контролера, так і системи управління в цілому.

Для аналізу моделей регуляторів та уточнення рішень з управління ними необхідно використовувати не одну, а кілька нечітких моделей об'єкта. Відомо, що погляди людини відображаються в моделі, заломлюючись через призму досвіду та накопичених практичних знань. Цілеспрямоване створення структури моделі й пов'язане з ним абстрагування від другорядних з точки зору мети параметрів «загрублюють» модель. Але разом з тим, розглядаючи різні аспекти мети дослідження і будуючи для кожного з них свою модель об'єкта (більш просту), що висвітлює визначену сторону, чи то впливу характеру робочого навантаження, чи внутрішніх параметрів силового виконавчого пристрою (ВП), чи алгоритму отримання достовірних параметрів і т.ін., можна одержати сукупність моделей. Така сукупність дає краще уявлення про проблему регулювання в цілому. Ускладнення моделі має відбуватися поступово та в кінцевому варіанті максимально відображати як модель системи управління, так і власне регулятора в цілому, а в максимальному наближенні враховувати і конкретний об'єкт.

На сьогоднішній день у роботах закордонних та вітчизняних спеціалістів представлені моделі систем управління з використанням нечіткої логіки, які призначені для вирішення технічних проблем керування різними технологічними об'єктами. Основні підходи та принципи побудови і інтеграції нечіткого логічного контролера НЛК (FLC) з класичними методами теорії автоматичного регулювання мають один і той же характер для різних прикладних моделей.

Тому розглянемо основні принципи організації та створення структур моделей нечітких систем регулювання напруги і шляхи імплантації НЛК у разі їх побудови.

Найбільш проста модель, яку можна рекомендувати для створення регуляторів, представлена на рис. 4.1. Система використовує сигнали датчиків від ВП (виробничого процесу, об'єкту) як вхідні для нечіткого логічного контролера. Представлена модель регулятора, незважаючи на її простоту, використовується досить не часто. Причиною цього є те, що поточні величини вихідних змінних залежать виключно від поточних значень вхідних і не враховують попередніх значень, окрім випадку, де жодне правило його роботи не активне і не визначені значення за умовчужанням. По суті, НЛК є статичним регулятором і призводить до статичної похибки у разі регулювання напруги.

Статичну похибку можна зменшити або повністю усунути шляхом введення додаткового НЛКд (рис. 4.2.). Останній, аналізуючи необхідні вихідні параметри та команди управління ВП, формує рішення щодо управління, яке призводить до коригування бази правил або вагових коефіцієнтів правил основного НЛК. Це, у свою чергу, призводить до зміни команди управління для ВП.

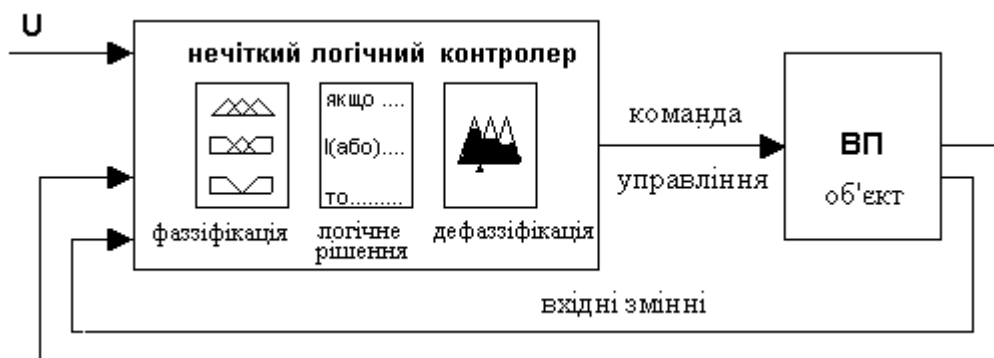


Рис. 4.1. Структурна схема моделі системи з НЛК

Для надання динаміки системі управління на основі НЛК динамічні функції мають забезпечуватися зовнішнім функціональним блоком. Такими блоками можуть виступати звичайні диференціальні та інтегральні елементи першого порядку, вихідні змінні яких є додатковими вхідними змінними НЛК, або виступати як самостійні.

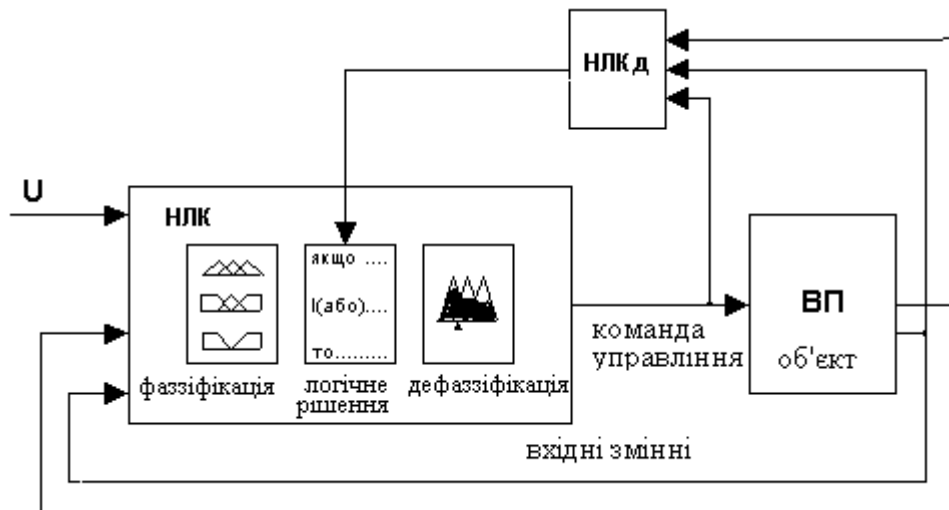


Рис. 4.2. Структурна схема нечіткої моделі з додатковим НЛК

На рис. 4.3 представлено модель нечіткого логічного контролера з пропорційним, інтегральним та диференціальним елементами. Сигнали неузгодження, тобто відхилення від заданого значення параметра, та похідної від сигналу використовуються як вхідні для НЛК. Вихідний сигнал контролера є вхідним для інтегратора, який формує командний сигнал для ВП. Подібні структури до представленої моделі в закордонній літературі зустрічаються під назвою нечіткого пропорційно–інтегрального контролера.



Рис. 4.3. Структурна схема моделі перетворювача напруги з нечітким пропорційно–інтегральним контролером

Поясненням цьому є те, що НЛК є нелінійним, і на основі вхідних сигналів він, насамперед, формує стратегію управління відповідно до своєї передатної характеристики. Його вихідний сигнал, пропорційний відповідно до характеристики вхідним параметрам після інтегратора, є командним для ВП. Тому модель з такою структурою можна називати нечітким пропорційно–інтегральним контролером (НПІК).

Однією з позитивних характеристик НПК є те, що його вихідний сигнал не має початкового значення. Нечіткий логічний контролер на основі правил бази знань оцінює неузгодження між сигналом зворотного зв'язку та заданим значенням параметра. Саме тенденція зміни неузгодження є критерієм того, чи збільшувати, чи зменшувати вихідний сигнал управління, а абсолютне значення сигналу команди не має значення.

Друга позитивна характеристика нечіткого пропорційно–інтегрального контролера – це його легке розширення щодо контролю додаткових параметрів для підвищення якості регулювання та стабільності.

Наприклад, за регулювання або ж стабілізації напруги в разі збурень зміна внутрішніх параметрів ВП чи потужності споживача може призвести до зниження напруги, яке не враховується НЛК. Але, якщо ввести додатково датчик контролю за відхиленням напруги на виході перетворювача, то такого зниження можна позбутися. Або інший приклад: у разі регулювання за відхиленням можна ввести додатковий датчик контролю величини струму та його фазового зсуву відносно напруги, що дасть можливість контролювати потужність споживача та характер навантаження. Останній чинник зумовлює зсув діапазону регулювання або стабілізації в бік зменшення значень меж у разі ємнісного характеру навантаження (наприклад, з 140–260 В до 130–250 В) і в бік збільшення значень меж у випадку індуктивного характеру навантаження. Тобто введення додаткових параметрів для обробки НЛК, яке не призводить до суттєвих апаратних витрат, дає змогу контролеру формувати адекватний ситуації сигнал управління та завчасно попереджати наслідки відхилення меж діапазону напруги, тобто зміни напруги на виході перетворювача.

На рис. 4.4 представлено структурну схему моделі нечіткого пропорційно–інтегрально–диференціального контролера (НПІДК), що використовується у разі регулювання напруги шляхом комутації силових ключів на відпайках трансформатора ТКВС і, одночасно, широтно–імпульсної модуляції напруги на частоті мережі (фазове управління) з використанням силових ключів на додатковій відпайці. В моделі нечіткий логічний контролер використовується для формування стратегії визначення необхідних значень параметра регулювання для класичних ПД контролерів (регуляторів), один з яких виконує грубе регулювання параметра, а другий – точне. За кожний момент часу дискретизації НЛК виставляє необхідну уставку грубого та точного регулювання. Така модель дає змогу організувати за допомогою НЛК багаторівневе регулювання одного чи декількох параметрів електричної мережі. Перевага від використання в наведеній моделі НЛК полягає в тому, що багаторівневе регулювання набагато легше формалізувати за допомогою правил на основі накопичених знань, ніж створювати математичну модель.

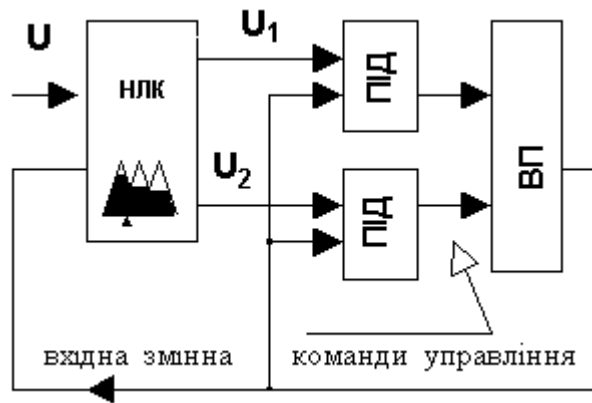


Рис. 4.4. Структурна схема моделі нечіткого пропорційно-інтегрально-диференціального контролера

Для реалізації автоматичної системи управління з підвищеною точністю доцільно використовувати модель, в якій НЛК працює в паралельному режимі з класичним ПІД контролером (рис. 4.5). У цьому випадку вихідні сигнали НЛК та ПІД контролера складаються. Тобто за нормальних умов, коли вихідний сигнал НЛК дорівнює нулю, працює класичний регулятор. У разі, коли НЛК виявляє відхилення від експлуатаційних режимів, то виконується коригування сигналу управління ВП. У представленій моделі контролюватися може не тільки вихідний параметр, але і сигнал управління ПІД контролера (показано штриховою лінією на рис.4.5).

На рис. 4.6 показано структурну схему моделі з більш «тісною» інтеграцією НЛК та класичного ПІД регулятора. Нечіткий контролер інтерпретує реакцію на виході моделі і здійснює налагодження ПІД регулятора шляхом установки нових значень коефіцієнтів пропорційності, інтегрування та диференціювання. Таким чином, реалізується модель самоналагоджувальної нестационарної лінійної системи, тобто лінійної системи зі змінними параметрами. Подібного типу структури використовуються в складних адаптивних системах управління.

У випадку створення автоматичної системи для відповідальних споживачів можливе використання НЛК з попередньою обробкою вихідного параметра. Наприклад, у системі регулювання напруги, виконання опрацювання сигналу на предмет аналізу спектра гармонічного складу напруги для прийняття рішення щодо коригування нелінійного спотворення форми кривої напруги мережі. На рис. 4.7 представлено структуру моделі, яка може використовуватися для дослідження та покращення якості напруги мережі (кондиціонування).

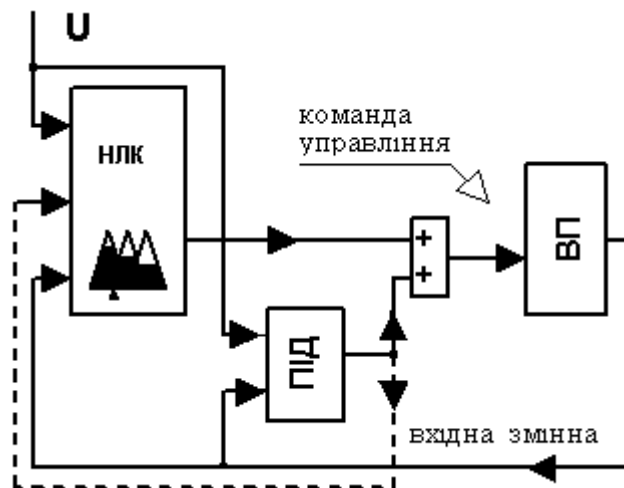


Рис. 4.5. Структурна схема моделі з паралельно ввімкненими НЛК та класичним пропорційно–інтегрально–диференціальним регулятором

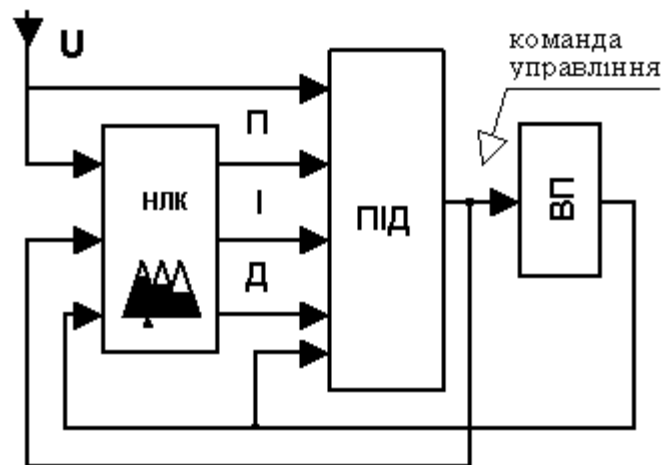


Рис. 4.6. Структурна схема моделі з послідовно ввімкненими НЛК та класичним пропорційно–інтегрально–диференціальним регулятором

У всіх запропонованих варіантах моделей необхідно враховувати, що не тільки структура вибраної моделі та кількість вхідних параметрів регулятора, які використовуються для аналізу в НЛК, визначають якість вихідного параметра, а також і внутрішня організація нечіткого контролера з оптимально розміщеними нечіткими множинами, тобто лінгвістичними змінними на базових фізичних змінних (параметрах) та правильне введення їх обмежень у вигляді функцій належності. Окрім цього на якість регулювання суттєво впливає правильне формування правил у базі знань контролера.

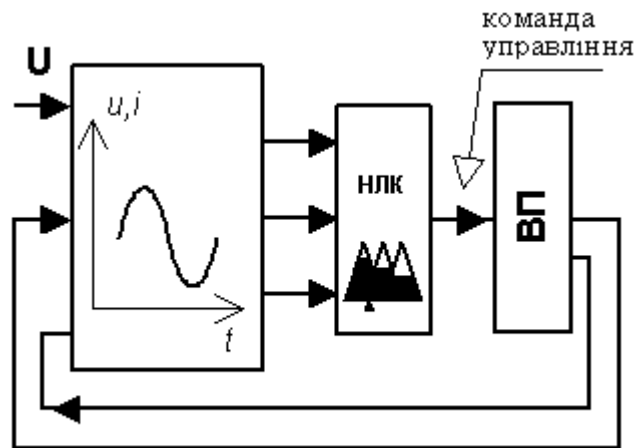


Рис. 4.7. Структурна схема моделі з самостійним формуванням НЛК базової вхідної змінної

Правила бази знань контролера мають максимально відтворювати весь спектр взаємодії «вхід–вихід», адекватно реагувати на всі можливі ситуації та спиратися на надбаний досвід у функціонуванні класичних систем регулювання і асоціативних відображень. Правила, що не активуються в процесі прийняття рішення мають виключатися з бази знань для покращення реакції контролера. Остаточне формування НЛК має базуватися на всебічному дослідженні системи «НЛК – Об’єкт» на візуальній імітаційній моделі.

Практичне застосування тієї чи іншої структурної схеми моделі має ґрунтуватися на всебічному аналізі режимів об’єкта та поставлених задач.

4.3 Методологія нечіткої логіки

Один із фундаментальних принципів сучасної науки полягає в тому, що явище не можна вважати зрозумілим, поки воно не описане за допомогою кількісних характеристик.

Зауваження

У 1883 р. Лорд Кельвін писав: «У фізичній науці при вивченні будь-якого об’єкта перший і найбільш істотний крок полягає в тому, щоб знайти принципи числової оцінки й практичні методи виміру деякої якості, властивого цьому об’єкту. Я часто говорю, що коли ви можете виміряти те, про що ви говорите, і виразити це в числах, ви вже знаєте дещо про це, але коли ви не можете виміряти це, коли ви не можете виразити це в числах, ваші знання недостатні й незадовільні: вони можуть бути початком знань, але ви лише злегка, подумки, просунулися до наукового розуміння питання, про що б не йшла мова».

З цього погляду багато чого з того, що становить сутність наукового знання, можна розглядати як сукупність принципів і методів, необхідних для

конструювання математичних моделей різних систем, що дають змогу одержувати кількісну інформацію про їхню поведінку. Поява обчислювальних машин викликала широке поширення кількісних числових методів у багатьох сферах людського знання. Безперечно, обчислювальні машини виявилися високоефективними у разі роботи з *механістичними* системами, тобто із системами, поведінка яких визначається законами механіки, фізики, хімії й електромагнетизму. На жаль, те ж саме не можна сказати про *гуманістичні* системи, які, принаймні дотепер, виявляли досить стійкий опір математичному аналізу і моделюванню із застосуванням ЕОМ.

Зауваження

Гуманістичними називаються такі системи, на поведінку яких сильний вплив виявляють судження, сприйняття або емоції людини. Приклади гуманістичних систем: економічні системи, політичні системи, правові системи, загальноосвітні системи та ін. Сама людина (індивідум) і його розумові процеси також можуть розглядатися як гуманістичні системи.

Насправді, багатьма допускається, що використання обчислювальних машин не пролило багато світла на основні проблеми, що виникають у філософії, психології, літературі, праві, політиці, соціології й в інших гуманітарних галузях знань. Обчислювальні машини не просунули нас скільки-небудь значно в розумінні процесу мислення, за винятком, можливо, деяких прикладів з сфери штучного інтелекту й суміжних сфер.

Таким чином, можливо саме з цієї причини *звичайні методи аналізу систем і моделювання на ЕОМ засновані на точній обробці числових даних по суті не здатні охопити величезну складність процесів людського мислення та прийняття рішень*. Звідси напрошується висновок про те, що для одержання правильних висновків про поведінку складних нелінійних та нестационарних систем, очевидно, необхідно відмовитися від високих стандартів точності та чіткості, які ми, як правило, очікуємо у випадку математичного аналізу класичних технічних (механістичних) систем.

Жертвуючи точністю перед лицем приголомшливої складності процесів та систем, природно вивчити можливість використання так званих лінгвістичних змінних, тобто змінних, значеннями яких є не числа, а слова або вирази (предикати) в природній або формальній мові.

Лінгвістичною ми називаємо змінну, значеннями якої є слова або вирази природньої або штучної мови.

Наприклад, *Вік* – лінгвістична змінна, якщо вона набуває лінгвістичних, а не числових значень, тобто значення *молодий, не молодий, дуже молодий, цілком молодий, старий, не дуже старий і не дуже молодий*, а не 20, 21, 22, 80 та ін.

Більш адекватно лінгвістична змінна описується набором $\{X, T(X), U, G, M\}$, в якому X – назва цієї змінної; $T(X)$ – терм–множина X , тобто сукупність її лінгвістичних значень; U – універсальна множина; G – синтаксичне правило, що породжує терми множини $T(X)$; M – семантичне правило, яке кожному лінгвістичному значенню X ставить у відповідність його зміст $M(X)$, причому $M(X)$ позначає нечітку підмножину множини U .

Зміст лінгвістичного значення X характеризується функцією належності: $U \rightarrow [0,1]$, яка кожному елементу $u \in U$ ставить у відповідність значення ступеня належності цього елемента до X . Так, наприклад, належність віку 27 років до значення *молодий* може дорівнювати 0.7, а віку 35 років – 0.2. Призначення семантичного правила – пов'язати належність так званих первинних термів у збірному лінгвістичному значенні з сумісністю збірного значення, наприклад, первинних термів *молодий* і *старий* у збірному лінгвістичному значенні *не дуже молодий і не дуже старий*. Невизначеності, такі як *дуже*, *цілком*, *надзвичайно* і ін., а також сполучники *і*, *або* розуміються у цьому разі як нелінійні оператори, що перетворюють зміст відповідних термів.

За допомогою лінгвістичних змінних можна приблизно описувати явища, які настільки складні або погано визначені, що не піддаються опису в загальноприйнятих кількісних термінах. Зокрема, якщо розуміти *Істинність* як лінгвістичну змінну зі значеннями *істинно*, *дуже істинно*, *зовсім істинно*, *не дуже істинно*, *не істинно* та ін., то ми приходимо до так званої нечіткої логіки. На таку логіку можуть опиратися наближені (тобто не повною мірою визначені) міркування, і вона у такий спосіб може служити більш реалістичною схемою людських міркувань, ніж традиційна двозначна логіка.

Найбільш ефективно використання лінгвістичного підходу знаходиться у сфері гуманістичних систем, особливо в таких сферах, як штучний інтелект, процеси прийняття рішень людиною, розпізнавання образів, психологія, право, медична діагностика, пошук інформації, економіка, і пов'язаних з ними сферах.

Використання слів і виразів (предикатів), а не чисел мотивоване тим, що лінгвістичний опис, як правило, менш конкретний, ніж опис числами. Наприклад, фраза «Напруга низька» менш конкретна, ніж фраза «напруга 180 В». У цьому сенсі слово *низька* можна розглядати як *лінгвістичне значення* змінної *Напруга*, маючи на увазі у цьому разі, що лінгвістичне значення відіграє таку ж роль, як і числове значення 180, але є менш точним і відповідно менш інформативним. Те ж саме можна сказати про лінгвістичні значення *дуже низька*, *не низька*, *надзвичайно низька*, *не дуже низька* та ін., якщо їх зіставити з числовими значеннями 150, 215, 140, 200 В, у діапазоні номінальної напруги мережі 220 В.

Якщо значення числової змінної зображують графічно точками на площині, то значення лінгвістичної змінної можна зобразити графічно як відрізки (площинки) з нечітко обкресленими межами. Саме завдяки такій

інтерпретації – використанню відрізків, а не точок – лінгвістичні змінні можуть служити засобом наближеного опису явищ, які настільки складні або некоректно визначені, що не піддаються точному опису.

Сукупність значень лінгвістичної змінної складає *терм-множина* цієї змінної. Ця множина може мати нескінченне число елементів. Наприклад, терм-множину лінгвістичної змінної **Напруга** можна записати так:

U(Напруга) = *низька + не низька + дуже низька + не дуже низька + дуже дуже низька + ... + висока + не висока + дуже висока + не дуже висока + не дуже висока і не дуже низька + ... + середня + не середня + ... + не висока і не середня + ... + надзвичайно висока.*

Знак + позначає тут об'єднання, а не арифметичну суму. Аналогічно терм-множина лінгвістичної змінної **Зовнішність** можна записати в такий спосіб:

T(Зовнішність) = *прекрасна + гарненька + миловидна + гарна + приваблива + не прекрасна + дуже гарненька + дуже дуже гарна + більш-менш гарненька + досить гарненька + досить гарна + безумовно гарна + не дуже приваблива, але й не дуже неприваблива + ...*

Наприклад, у випадку лінгвістичної змінної *напруга* числова змінна напруга, яка набуває значення 0, 1, 2, 3, ..., 380 В, є так званою *базовою змінною (універсумом)* лінгвістичної змінної *напруга*. У цьому разі таке, наприклад, лінгвістичне значення, *низька*, можна інтерпретувати як назва деякого *нечіткого обмеження* на значення базової змінної. Саме це обмеження ми і вважаємо змістом лінгвістичного значення *низька*. Графічна інтерпретація лінгвістичної змінної *напруга* представлена на рис. 4.8.

Нечітке обмеження на значення базової змінної характеризується *функцією належності*, яка кожному значенню базової змінної ставить у відповідність число з інтервалу [0,1], що символізує сумісність цього значення з нечітким обмеженням. Наприклад, значення функції належності таких числових значень змінної *вік*, як 22, 28 і 35 з нечітким обмеженням *молодий*, можуть бути 1, 0,7 і 0,2 відповідно.

У цьому випадку зміст лінгвістичної величини *молодий* можна представити кривою (рис. 4.9), що є графіком функції належності лінгвістичного значення *молодий* відносно базової змінної *вік*.

Звичайна інтерпретація висловлення «напруга низька» полягає в тому, що ця напруга належить діапазону низьких значень. Однак, беручи до уваги, що діапазон низька – нечітка множина, тобто що не існує різкого переходу від низької до високої, твердження про те, що напруга належить діапазону низьких, не відповідає точному математичному визначенню поняття «належить». Концепція лінгвістичної змінної дає змогу обійти ці труднощі в такий спосіб.

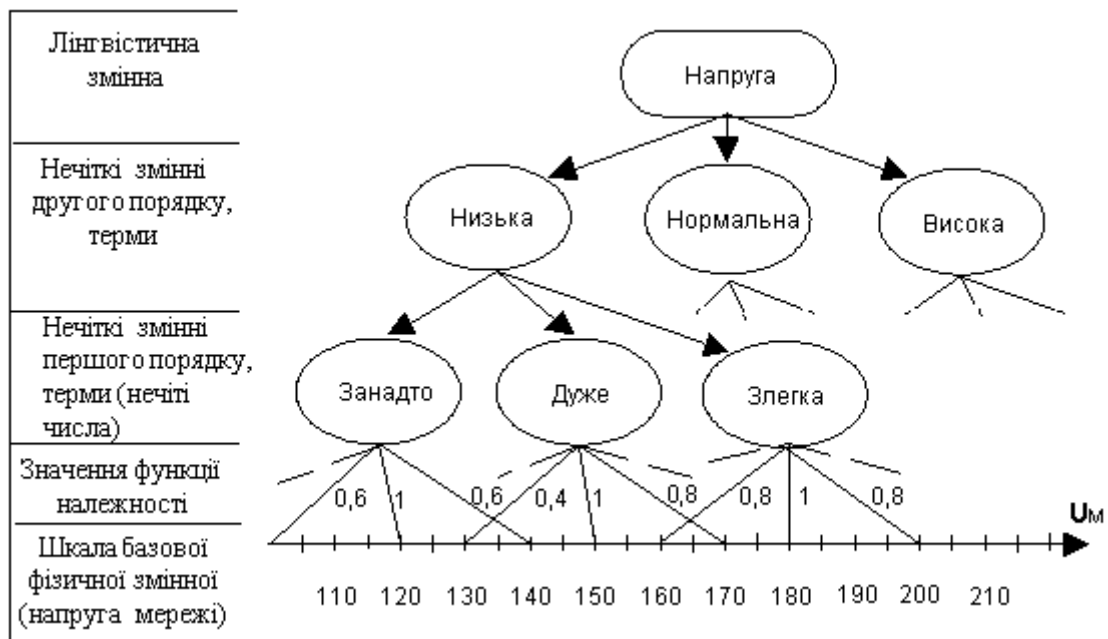


Рис. 4.8. Ієрархічна структура лінгвістичної змінної *напруга*

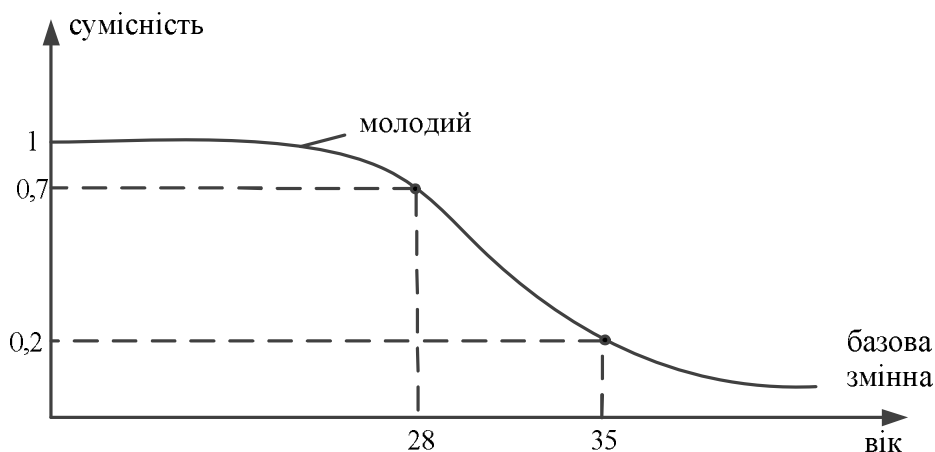


Рис. 4.9. Функція належності нечіткої множини *молодий* лінгвістичної змінної *вік*

Лінгвістична змінна має лінгвістичну *структуру* в тому розумінні, що вона пов'язана із двома правилами.

Перше – *синтаксичне правило* – визначає спосіб породження лінгвістичних значень, що належать до терм-множини цієї змінної. У цьому разі ми будемо звичайно припускати, що елементи терм-множини породжуються безконтекстною граматикою.

Друге – *семантичне правило* – визначає спосіб обчислення змісту будь-якої лінгвістичної змінної. У зв'язку з цим зазначимо, що типові значення лінгвістичної змінної, наприклад *не дуже молодий* і *не дуже старий*, містять у

собі те, що можна було б назвати *первинними термами*, наприклад *молодий* і *старий*, зміст яких суб'єктивний і залежить від контексту. Передбачається, що зміст таких термів визначений заздалегідь.

Крім первинних термів лінгвістичне значення може містити в собі такі сполучники, як *і*, *або*, ..., *ні*; заперечення *не*; такі невизначеності (прислівники), як *дуже*, *більш–менш*, *зовсім*, *безумовно*, *надзвичайно* та ін.

Сполучники, невизначеності і заперечення можна трактувати як оператори, які видозмінюють зміст первинних термів особливим, незалежним від контексту чином. Так, якщо функція сумісності лінгвістичного значення *молодий* зображується кривою, показаною на рис. 4.10, то зміст лінгвістичного значення *дуже молодий* може бути отриманий зведенням у квадрат значень функції належності лінгвістичного значення *молодий*. Зміст лінгвістичного значення *не молодий* можна одержати, віднімаючи від одиниці значення функції належності *молодий* (рис. 4.10).

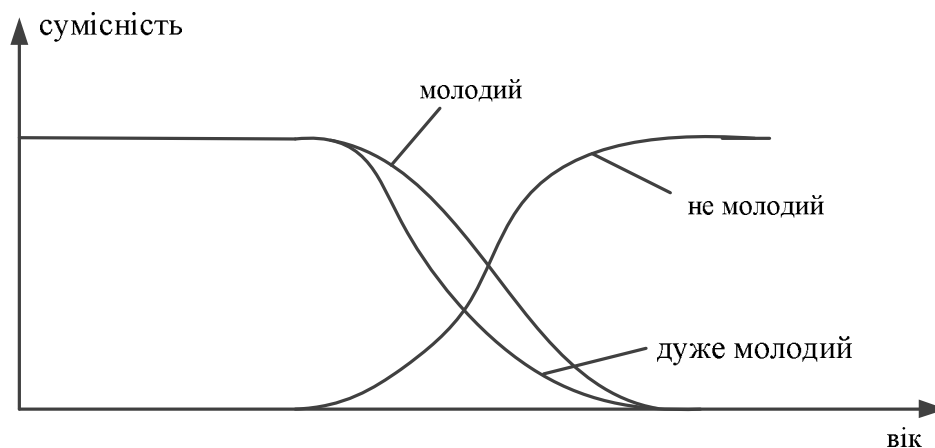


Рис. 4.10. Функції належності нечітких множин *молодий*, *не молодий*, *дуже молодий*

Лінгвістичний підхід не є за своєю природою цілком якісним. Вірніше, обчислення відбуваються «розумово» (віртуально), де використовується *лінгвістичне наближення* для перетворення числових значень у слова.

4.4 Стохастична і лінгвістична невизначеності

Для більш детального розгляду нечітких лінгвістичних змінних необхідно уточнити, чи є нечіткість різновидом імовірності, чи вона має якийсь самостійний зміст?

Історично вивченням і розробкою моделей, що враховують невизначеність того або іншого виду, займається багато математичних дисциплін, такі як теорія ймовірностей, теорія інформації, математична

статистика, теорія ігор, теорія масового обслуговування та теорія нечітких множин. Класифікуємо типи невизначеностей для виявлення відмінностей між ними.

Стохастична невизначеність має місце в ситуаціях, коли деяка обґрунтовано описана подія може відбутися, а може не відбутися. У цьому випадку із часом ступінь невизначеності, пов'язана з цією подією, може змінюватися.

Розглянемо наступний вираз: *«Імовірність того, що при виникненні грози над підстанцією, захищеною чотирма блискавковідводами, буде вражено один з відводів, дорівнює 0,25»*.

У цьому виразі неявно передбачається, що форма відводів та їх висота над поверхнею землі є однаковою. По закінченні деякого часу невизначеність зникає, оскільки гроза закінчиться.

Таким чином, розглянутий вираз має сенс тільки стосовно події в майбутньому.

Історично теорія ймовірностей була першою математичною дисципліною для виявлення невизначеності в математичних моделях. Із цієї причини будь-яка невизначеність тривалий час вважалася стохастичною за своєю природою і наділялася, іноді штучно, властивостями випадкової невизначеності. Що стосується імовірнісного процесу, результат будь-якої окремої реалізації якого є винятково випадком, передбачити послідовність подій просто неможливо. Для імовірнісних процесів виявляється можливим лише точний опис статистичних оцінок деяких узагальнених характеристик цього процесу.

Розглянемо інший вираз:

«Імовірність того, що завтра піде дощ, дорівнює 0,7».

У цьому виразі неявно передбачається, що подія *«підє дощ»* ґрунтовно описана. Проте зовсім очевидно, що ця подія недостатньо визначена: не ясно, чи то дощ буде йти цілий день, чи то дощ буде йти 70% від часу наступної доби? Більше того, чи можливо вважати дощем дрібний дощ або тільки зливу?

Розглянемо інший вираз: *«Імовірність того, що завтра о 17 годині буде максимум добового навантаження, дорівнює 0,6»*.

У цьому виразі неявно передбачається, що подія *«максимум добового навантаження»* добре описана. Проте зовсім очевидно, що ця подія недостатньо визначена: не ясно, як довго триватиме максимум навантаження, чи то 30 хвилин, чи декілька годин? Більше того, чи можливо вважати однохвилинний скачок навантаження як максимум?

Таким чином, у разі очевидності розглянутих виразів за більш детального їх аналізу ми виявляємо деякий інший тип невизначеності, який змістовно відрізняється від стохастичної. Ця невизначеність скоріше відноситься до лінгвістичного опису ситуації або події, а не до кількісної оцінки того, відбудеться ця подія в майбутньому або не відбудеться.

Лінгвістична невизначеність пов'язана з неточністю звичайної людської мови, з нею ми постійно зустрічаємося в повсякденнім житті. Досить розглянути фрази типу «суттєве навантаження», «високі люди», «гарячі пиріжки», «гарна особа», «гарний автомобіль», «стала валюта», «важкий день», щоб зрозуміти, що навряд чи можливо дати їм точні кількісні визначення.

Дійсно, досвідчені й недосвідчені люди будуть мати свої власні думки про те, яких людей слід вважати інтелегентними. Більше того, якщо ми формально встановимо – інтелегентні люди це ті, що відвідують театр, то чи будуть люди що відвідують кінотеатр інтелегентними чи ні? Контекст фраз теж має значення, оскільки оцінка інтелегентних людей в селі і у місті буде різною.

Для вивчення подібних суб'єктивних оцінок призначена окрема наука – психолінгвістика. У рамках цієї науки прийнято вважати, що в розглянутих фразах люди використовують слова в якості деяких суб'єктивних категорій. Ці суб'єктивні категорії дають нам можливість класифікувати об'єкти, які характеризуються такими властивостями, як «висота», «довжина», «вага», «температура», «колір». Навіть при тому, що більшість використовуваних категорій точно не визначена, люди можуть використовувати їх для досить адекватних оцінок і рішень, які засновано на обліку багатьох різних факторів.

У деяких випадках невизначеність, яка присутня в значенні тих або інших слів, свідомо використовується нами в розмові для додання їй додаткової гнучкості.

Для адекватного використання логіки, що притаманна людським міркуванням, у випадку вирішення технічних проблем необхідно розробити відповідну математичну модель лінгвістичної невизначеності. Саме з цією метою була розроблена нечітка логіка, яка дає змогу представити процеси прийняття рішень і оцінки ситуацій людиною в деякій алгоритмічній формі. Хоча можливості людського мислення і фантазії безмежні, межі того, що дає змогу моделювати нечітка логіка, існують.

Яким чином люди можуть міркувати щодо реальних систем, коли кінцевий опис реальної системи часто вимагає більш детальних даних, ніж людина може одержати та інтерпретувати? Відповідь полягає в тому, що люди мають здатність міркувати приблизно, можливість, якої комп'ютери не мають. У разі спілкування людей використання фраз типу «висока людина» і «високооплачувана робота» не призводить до виникнення концептуальних проблем, оскільки передає семантично зрозумілу інформацію особам, які беруть участь у розмові. За необхідності завжди можна уточнити використовувані суб'єктивні категорії.

На той же час комп'ютери або мікропроцесори використовують у своїй роботі винятково бінарну логіку. Для розуміння відповідних фраз штучним інтелектом (комп'ютером) необхідно, щоб конкретне значення напруги або сили струму зрівнювалося із заданим граничним значенням для розглянутих формальних категорій «висока напруга» і «слабкий струм».

Основна властивість теорії нечітких множин полягає в можливості використовувати лінгвістичні змінні замість кількісних, нечітку логіку замість бінарної логіки Буля, для формального визначення подібних неточних суб'єктивних категорій.

У випадку розглядання складної системи люди міркують щодо її структури й поведінки приблизно або неточно. Тим самим досягається деяке універсальне розуміння змісту проблеми. На щастя, ця спільність і неточність, що здобуваються у формі досвіду із часом, найчастіше виявляються достатніми для людського розуміння складних явищ і адекватного прийняття рішень у побутових ситуаціях. Саме в рамках теорії нечітких множин виявляється можливим включити в опис проблеми цей досвід і інтуїцію окремої людини.

Розглянуті вище приклади висловлювання ілюструють той факт, що стохастична та лінгвістична невизначеності мають різний характер. Стохастична невизначеність має відношення з невизначеністю того, чи відбудеться деяка описана подія в майбутньому, а теорія ймовірностей дає змогу дати на це питання ту або іншу відповідь.

Навпаки, лінгвістична невизначеність пов'язана з неточністю опису самої ситуації або події незалежно від часу їх розгляду.

Теорія ймовірності не може використовуватися для вирішення подібних проблем, оскільки визначення про суб'єктивні категорії, що присутні у процесах мислення людини, повною мірою не узгодяться з її аксіомами.

Так, поняття нечіткої множини здатне забезпечити нас адекватною інформацією щодо недостатньо повного опису тих або інших ситуацій. По суті, цей підхід найбільше застосуємо для вирішення таких проблем, у яких невизначеність характеризується відсутністю певних критеріїв, що дають змогу однозначно судити про належність елементів тому або іншому класу. Саме в цьому виявляється відмінність між нечіткістю і випадковістю. У той же час нечіткі моделі не є заміною моделей, розроблених у теорії ймовірностей. Кожна чітка множина є нечіткою, але зворотне твердження неможливе. Тому нечіткі моделі узагальнюють традиційні й більш знайомі нам математичні моделі. Ефективність же моделі проявляється в її здатності адекватно розв'язати ту або іншу конкретну проблему.

Як правило, складна проблема тією чи іншою мірою пов'язана з невизначеністю. Мистецтво і професіоналізм системного аналітика саме й проявляються в тому, щоб запропонувати для її рішення таку модель, яка найбільше адекватно враховує той або інший тип невизначеності. Досягнуті останнім часом вражаючі успіхи у використанні нечітких технологій для рішення найрізноманітніших практичних завдань дають змогу стверджувати, що нечітке моделювання реальних складних систем – ефективна альтернатива традиційним математичним моделям і методам.

4.5 Математичне визначення та характеристики чіткої і нечіткої множин

Розглянемо основні терміни і визначення теорії чітких (crisp set) множин. Великими літерами (наприклад, U) позначається сукупність об'єктів, а малими (наприклад, u) – окремі структурні елементи. У цьому разі справедливий запис $U = \{u\}$. Фігурні дужки означають сукупність об'єктів. Сукупність U називається повним простором або *універсумом*. Окремі структурні об'єкти є просто елементами. Той факт, що u є елементом U , позначається у такий спосіб: $u \in U$.

У повному просторі U визначимо чітку множину. У якості назв (термінів) множин використаємо великі літери букви A, B, C . Наприклад, нехай повна множина певних значень напруги складається з десятих числових значень $U = \{220, 221, 222, 223, 224, 225, 226, 227, 228, 229\}$, тоді множину A парних чисел можна представити як $A = \{220, 222, 224, 226, 228\}$.

Число структурних елементів (кількість) називається *потужністю* множини і позначається $\#$. У зазначених вище прикладах $\#U = 10$, $\#A = 5$. У випадку $\#A = 1$ A називається синглетоном. Множина з кінцевим $\#$ – називається кінцевою множиною, всі елементи в такій множині можна записати так, як представлені вище множини U та A . У випадку натуральних або дійсних чисел, тобто нескінченних множин, цього зробити не можна. Часто використовують засіб запису множини, коли праворуч від вертикальної риски записують усі властивості множини. Наприклад, множину A можна записати у вигляді

$$A = \{u \mid u - \text{парне число більше } 220, \text{ але менше } 229\}.$$

Окрім представлених понять чіткої множини існує засіб визначення за допомогою характеристичної функції. Характеристична функція (хі) χ_A , що визначає множину A в повному просторі, є собою відображенням, для якого U є областю визначення, а $\{0, 1\}$ (двозначна множина з 0 і 1) є областю значень:

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

У цьому разі $\chi_A(u) = 1$, якщо елемент u задовольняє властивостям A , і $\chi_A(u) = 0$, якщо не задовольняє. У повному просторі U можна розглядати різноманітні множини, наприклад, A з деякими властивостями і B з іншими властивостями. Об'єднання всіх таких множин називається степеневою множиною і позначається 2^U . Наприклад, нехай $U = \{a, b, c\}$, тоді степеневу множину можна записати у вигляді $2^U = \{0, \{a\}, \{b\}, \{c\}, \{a, b\}, \{b, c\}, \{c, a\}, U\}$, де 0 – це спеціальна множина, у котрій немає елементів, вона називається

порожньою множиною. Її характеристична функція $\chi_0(u)=0$ для $\forall u \in U$. Тут $\forall$ називається квантором загальності, його можна читати словом «усіх». (Крім нього є квантор існування $\exists$, тобто «існує. . .».) Ці квантори часто використовуються в логіці та штучному інтелекті. На відміну від порожньої множини характеристична функція повної множини U має вигляд $\chi_u(U)=1$ для $\forall u \in U$.

Розглянемо основні операції над множинами. Насамперед, відношення вкладення множин: якщо елементи A обов'язково є елементами B , то A називається підмножиною B (або B – надмножиною A). Це відношення позначається як $A \subset B$ і воно справедливе також, коли $A = B$. Якщо $A \subset B$, але $A \neq B$, то A називається власною підмножиною B . Відношення $A \subset B$ через характеристичну функцію записується таким виразом: $\chi_A(u) \leq \chi_B(u)$ для $\forall u \in U$. Для відношення вкладення множин характерні три властивості:

- 1) рефлексивність $\forall A \in 2^U, A \subset A$;
- 2) асиметричність $\forall A, B \in 2^U, A \subset B, B \subset A \rightarrow A = B$;
- 3) транзитивність $\forall A, B, C \in 2^U, A \subset B, B \subset C \rightarrow A \subset C$.

Для відношення вкладення множин у випадку, коли $A \subset B$ чи $B \subset A$, можна сказати, що $(2^U, \subset)$ утворює частково упорядковану множину. Для таких множин становлять інтерес унарні та бінарні операції. Це три основні операції: доповнення множин (наприклад, $\bar{A}$); перетину множин A і B ($A \cap B$); об'єднання множин A і B ($A \cup B$). За допомогою характеристичних функцій ці операції можна визначити так:

$$\chi_{\bar{A}}(u) = 1 - \chi_A(u) \quad \text{для } \forall u \in U,$$

$$\chi_{A \cap B}(u) = \chi_A(u) \wedge \chi_B(u) \quad \text{для } \forall u \in U,$$

$$\chi_{A \cup B}(u) = \chi_A(u) \vee \chi_B(u) \quad \text{для } \forall u \in U.$$

Тут символи $\wedge$ та $\vee$ називаються операціями мінімуму і максимуму, тобто взяття найменшого і найбільшого значень.

Нечітка множина (fuzzy set) є сукупністю елементів довільної природи, щодо яких не можна з повною визначеністю стверджувати – належить той або інший елемент розглянутої сукупності цій множині чи ні. Інакше кажучи, нечітка множина відрізняється від звичайної чіткої множини тим, що для всіх або частини її елементів не існує однозначної відповіді на запитання: «Належить, чи не належить той або інший елемент розглянутій нечіткій множині?»

Розглянемо математичне визначення нечіткої множини. Формально нечітка множина $\mathcal{A}$ визначається як множина упорядкованих пар або кортежів виду $\langle x, \mu_{\mathcal{A}}(x) \rangle$, де x є елементом деякої універсальної множини або універсума X , а $\mu_{\mathcal{A}}(x)$ – функція належності, яка ставить у відповідність кожному з елементів $x \in X$ деяке дійсне число з інтервалу $[0, 1]$, тобто ця функція визначається у формі відображення:

$$\mu_{\mathcal{A}} : X \rightarrow [0,1].$$

У цьому разі значення $\mu_{\mathcal{A}}(x)=1$ для деякого $x \in X$ означає, що елемент x *однозначно належить* нечіткій множині $\mathcal{A}$, а значення $\mu_{\mathcal{A}}(x)=0$ означає, що елемент x *однозначно не належить* нечіткій множині $\mathcal{A}$.

Формально кінцеву нечітку множину будемо записувати у вигляді $\mathcal{A} = \{ \langle x_1, \mu_{\mathcal{A}}(x_1) \rangle, \langle x_2, \mu_{\mathcal{A}}(x_2) \rangle, \dots, \langle x_n, \mu_{\mathcal{A}}(x_n) \rangle \}$, а в загальному випадку – у вигляді

$$\mathcal{A} = \{ \langle x, \mu_{\mathcal{A}}(x) \rangle \}.$$

Зауваження

Окрім прийнятих нами позначень кінцеві нечіткі множини часто записуються у формі: $A = \{(\mu_A(x_1), x_1), (\mu_A(x_2), x_2), \dots, (\mu_A(x_n), x_n))\}$, $A = \{x_1/\mu_A(x_1) + x_2/\mu_A(x_2) + \dots + x_n/\mu_A(x_n)\}$ або $A = \left\{ \frac{\mu_A(x_1)}{x_1} + \frac{\mu_A(x_2)}{x_2} + \dots + \frac{\mu_A(x_n)}{x_n} \right\}$. У цьому разі нахилена (слеш) та горизонтальна риски служать просто роздільником, а знак «+» позначає не арифметичну суму, а теоретико–множинне об'єднання окремих елементів. Бажаючи підкреслити або явно вказати, що множина A є нечіткою, багато авторів часто записують нечіткі множини зі знаком *тільда* «~» унизу або вгорі, тобто у формі $\underline{A}$ або $\tilde{A}$.

Порожня нечітка множина. У теорії нечітких множин зберігають свій зміст деякі спеціальні класичні множини. Так, наприклад, *порожня нечітка* множина або множина, яка не містить ніякого елемента, як і раніше позначається через $\emptyset$ і формально визначається як така нечітка множина, функція належності якої тотожно рівняється нулю для всіх без винятку елементів: $\mu_{\emptyset}=0$. У цьому зв'язку доречно згадати про те, що характеристична функція звичайної порожньої множини також тотожно рівняється нулю для будь–яких елементів: $\chi_{\emptyset} = 0$.

Універсум. Що стосується іншої спеціальної множини, то так званий *універсум*, позначуваний через X , вже був використаний вище як звичайна множина, яка містить у рамках деякого контексту всі можливі елементи. Формально зручно вважати, що функція належності універсума як нечіткої множини тотожно рівняється одиниці для всіх без винятку елементів: $\mu_x=1$. У цьому разі характеристична функція звичайної універсальної множини також тотожно рівняється одиниці для будь–яких елементів: $\chi_x = 1$.

Носій нечіткої множини. *Носієм* нечіткої множини $\mathcal{A}$ називається підмножина A_s , яка містить ті й тільки ті елементи універсума, для яких значення функції належності відповідної нечіткої множини відмінні від нуля. Математично носій нечіткої множини визначається такою умовою:

$$A_s = \{ x \in X \mid \mu_{\mathcal{A}}(x) > 0 \} \quad \forall x \in X.$$

Очевидно, порожня нечітка множина має порожній носій, оскільки $\mu_0=0$ для будь-якого його елемента. Носій універсума, розглянутого як нечітка множина, співпадає із самим універсумом.

Ядро нечіткої множини. Ядром нечіткої множини $\mathcal{A}$ називається така підмножина A_1 , елементи якої задовольняють умові

$$A_1 = \{x \in X \mid \mu_{\mathcal{A}}(x) = 1\}.$$

Наприклад, ядро нечіткої множини $\mathcal{A}$ «невелике натуральне число» дорівнює двоелементній множині $A_1 = \{1, 2\}$. Ядро нечіткої множини $\mathcal{B}$ «дійсне число, приблизно рівне нулю», рівняється одноелементній множині (singleton) $B_1 = \{0\}$. Нечітка множина $\mathcal{C}$ «велике дійсне число» має порожнє ядро.

На рис. 4.11 графічно функціями належності представлено ядро, носій і межі нечіткої множини, одна з яких є нормальною $\max[\mu_{\mathcal{A}}(x)] = 1$ (рис. 4.11,а), а інша – не нормальною $\max[\mu_{\mathcal{A}}(x)] < 1$ (рис. 4.11,б)

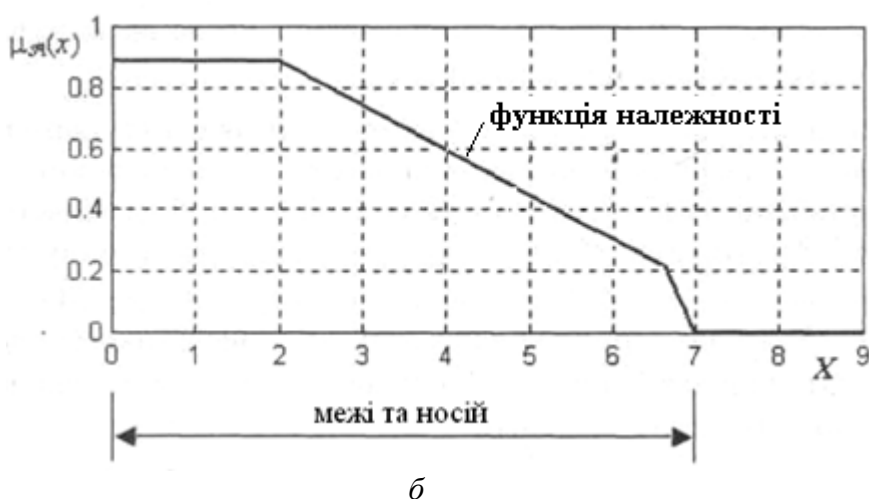
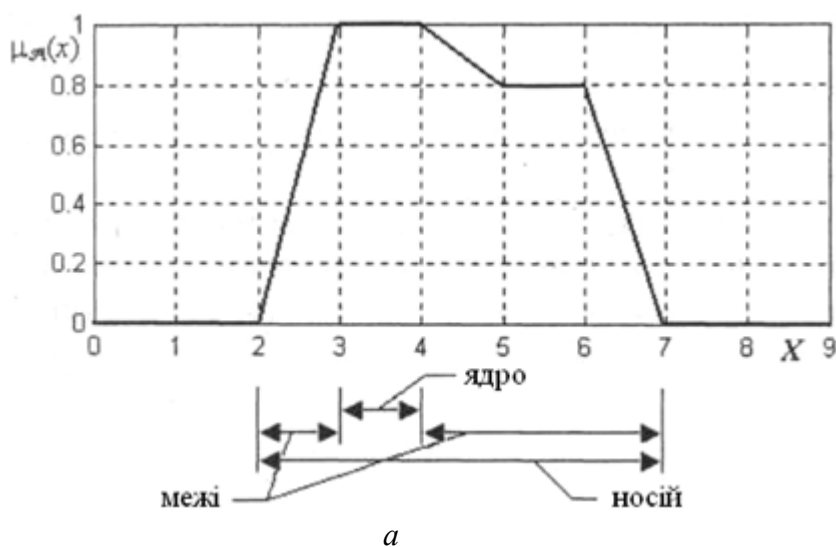


Рис. 4.11. Ядро, носій і межі нечітких множин

Межі нечіткої множини. Межами нечіткої множини називаються такі елементи універсума, для яких значення функції належності відмінні від 0 і 1. Інакше кажучи, межі нечіткої множини $\mathcal{A} = \{x, \mu_{\mathcal{A}}(x)\}$ включають ті й тільки ті елементи універсума $x \in X$, для яких виконується умова $0 < \mu_{\mathcal{A}}(x) < 1$.

Точка переходу нечіткої множини. Елементи нечіткої множини $y \in \mathcal{A}$, для яких виконується умова $\mu_{\mathcal{A}}(y) = 0.5$, називаються *точками переходу* цієї нечіткої множини $\mathcal{A}$ (за умови рівномірної щільності розміщення ФН).

4.6 Основні типи функцій належності

Нечітка множина A в повному просторі U визначається через функцію належності $\mu_A(u)$, де $u \in U$.

Графічно найпростіші типові функції належності відповідно до стандарту ІЕС 1131–7 представлені на рис. 4.12.

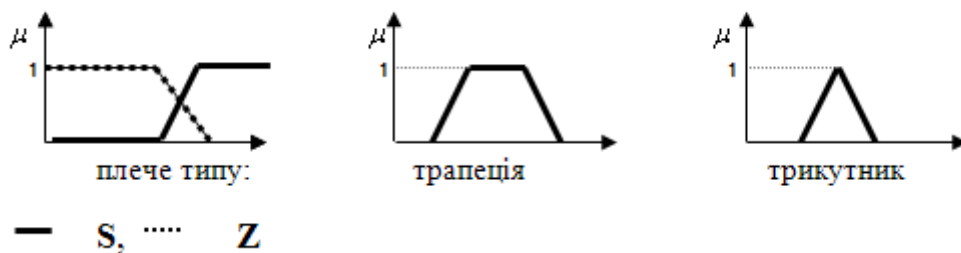


Рис. 4.12. Графіки типових функцій належності

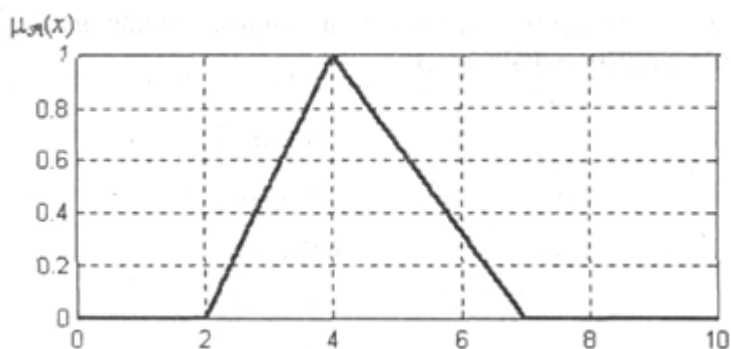
На рис. 4.13 представлені спеціальні графічні форми функції належності. Прямокутником описується функція належності, наприклад, інтервалу. Функції належності типу «синглетон» використовуються для альтернативного представлення вихідної змінної.



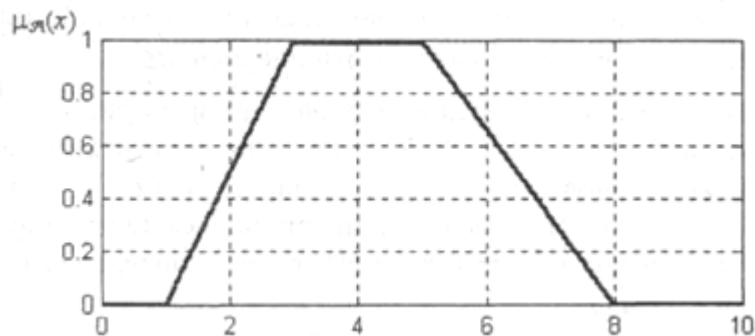
Рис. 4.13. Спеціальні графічні форми функції належності

Розглянемо кусково–лінійні функції належності.

Як кусково–лінійні функції належності розглянемо функції, які, як випливає з їхньої назви, складаються з відрізків прямих ліній, що утворюють неперервну або кусково–неперервну функцію. Найбільш характерним прикладом таких функцій є «трикутна» (рис. 4.14, а) і «трапецієподібна» (рис. 4.14, б) функції належності. У нашому випадку кожна із цих функцій задана на універсумі $X=[0, 10]$, у якості якого вибраний обмежений інтервал дійсних чисел. У загальному випадку вибір універсуму може бути довільним і не обмеженим жодними правилами.



а



б

Рис. 4.14. Графіки функцій належності: а – трикутної; б – трапецієподібної форми

Формальне визначення нечіткої множини не накладає ніяких обмежень на вибір конкретної функції належності для її наведення. Однак на практиці зручно використовувати ті з них, які допускають аналітичний опис у вигляді деякої простої математичної функції. Це спрощує не тільки відповідні числові розрахунки, але й скорочує обчислювальні ресурси, необхідні для зберігання окремих значень цих функцій належності.

Розглянемо в аналітичній формі трикутну та трапецієподібну ФН.

Трикутна функція належності в загальному випадку може бути задана аналітично таким виразом:

$$\mu_{\Delta}(x; a, b, c) = \begin{cases} \frac{x-a}{b-a}, & a \leq x \leq b \\ \frac{c-x}{c-b}, & b \leq x \leq c \\ 0, & c \leq x \end{cases},$$

де a, b, c – деякі числові параметри, що набувають довільних дійсних значень й упорядковані відношенням $a \leq b \leq c$.

Стосовно конкретної функції, зображеної на рис. 4.14, a , значення параметрів дорівнюють: $a=2, b=4, c=7$. Неважко помітити, параметри a і c характеризують основу трикутника, а параметр b – його вершину. Як можна помітити, ця функція належності характеризує нормальну випуклу унімодальну нечітку множину з носієм – інтервалом (a, c) , межами (a, b) і (b, c) , ядром $\{b\}$ і модою b .

Трапецієподібна функція належності в загальному випадку може бути задана аналітично таким виразом:

$$\mu_T(x; a, b, c, d) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ 1, & b \leq x \leq c \\ \frac{d-x}{d-c}, & c \leq x \leq d \\ 0, & d \leq x \end{cases},$$

де a, b, c, d – деякі числові параметри, що набувають довільних дійсних значень й упорядковані відношенням $a \leq b \leq c \leq d$.

Стосовно функції, зображеної на рис. 4.14, b , значення параметрів дорівнюють: $a=1, b=3, c=5, d=8$. Параметри a й d характеризують нижню основу трапеції, а параметри b і c – верхню основу трапеції. У цьому разі ця функція належності породжує нормальну випуклу нечітку множину із носієм – інтервалом (a, d) , межами (a, b) і (c, d) і ядром (b, c) .

Трикутна та трапецієподібна функції використовуються для представлення таких властивостей множин, які характеризують невизначеності типу: «приблизно рівно», «середня напруга», «тепла вода», «нормальний струм», «майже норма» та ін.

Покажемо в аналітичній формі Z-подібні та S-подібні функції належності. Z-подібні та S-подібні функції належності одержали свої назви за виглядом кривих, які представляють їхні графіки.

Перша з функцій цієї групи називається Z-подібною і у загальному випадку може бути задана аналітично таким виразом:

$$\mu_{Z_1}(x; a, b) = \begin{cases} 1, & x < a \\ \frac{1}{2} + \frac{1}{2} \cos\left(\frac{x-a}{b-a}\pi\right), & a \leq x \leq b \\ 0, & x > b \end{cases},$$

де a, b – деякі числові параметри, що набувають довільних дійсних значень й упорядковані відношенням $a < b$. Графік цієї функції для деякої нечіткої множини $\mathcal{A}$ і універсума $\mathbf{X}=[0, 10]$ зображений на рис. 4.15,а, у цьому разі значення параметрів відповідно дорівнюють $a=3, b=6$.

Сплайн-функція Z -подібної функції може бути також задана іншим виразом:

$$\mu_{Z_2}(x; a, b) = \begin{cases} 1, & x \leq a \\ 1 - 2\left(\frac{x-a}{b-a}\right)^2, & a < x \leq \frac{a+b}{2} \\ 2\left(\frac{b-x}{b-a}\right)^2, & \frac{a+b}{2} < x < b \\ 0, & b \leq x \end{cases},$$

де a, b – деякі числові параметри, що набувають довільних дійсних значень й упорядковані відношенням $a < b$. Графік цієї функції для деякої нечіткої множини $\mathcal{A}$ і універсума $\mathbf{X} = [0, 10]$ зображений на рис. 4.15,б, у цьому разі значення параметрів відповідно $a = 3, b = 6$.

Зауваження

Хоча на перший погляд відмінність між цими функціями ледь помітна, проте вона існує, у чому можна переконатися за допомогою представлення їх графіків на одному рис. 4.16.

Такі функції належності породжують нормальні випуклі нечіткі множини з ядром $(-\infty, a)$ і носієм $(-\infty, b)$. Вони використовуються для наведення таких властивостей нечітких множин, які характеризуються невизначеністю типу: «мала кількість», «невелике значення», «низька напруга», «низька собівартість продукції», «низький рівень цін або доходів», «низька процентна ставка» і багатьох інших. Загальним для всіх цих ситуацій є слабкий ступінь прояву тієї або іншої якісної або кількісної ознаки. Особливість нечіткого моделювання у цьому разі полягає у наведенні відповідних нечітких множин за допомогою незростаючих (монотонно спадаючих) функцій належності.

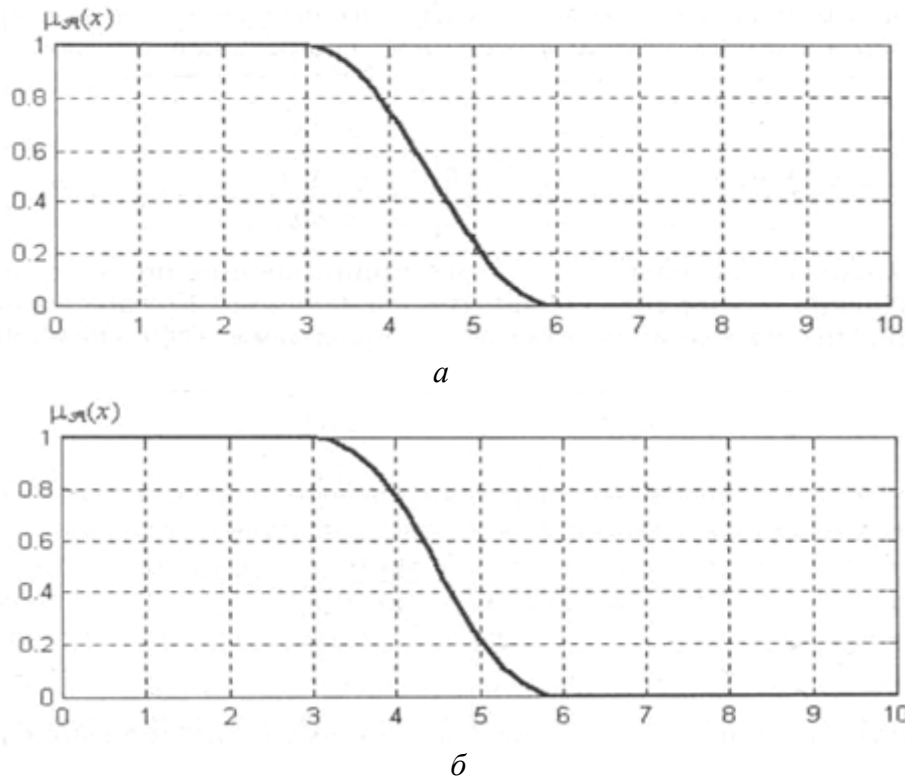


Рис. 4.15. Графіки Z-подібних функцій належності для значень параметрів $a=3$, $b=6$: $a - fz_1$; $б - fz_2$

S-подібну функцію належності у загальному випадку можна задати аналітично таким виразом:

$$\mu_{S_1}(x; a, b) = \begin{cases} 0, & x < a \\ \frac{1}{2} + \frac{1}{2} \cos\left(\frac{x-b}{b-a} \pi\right), & a \leq x \leq b \\ 1, & x > b \end{cases},$$

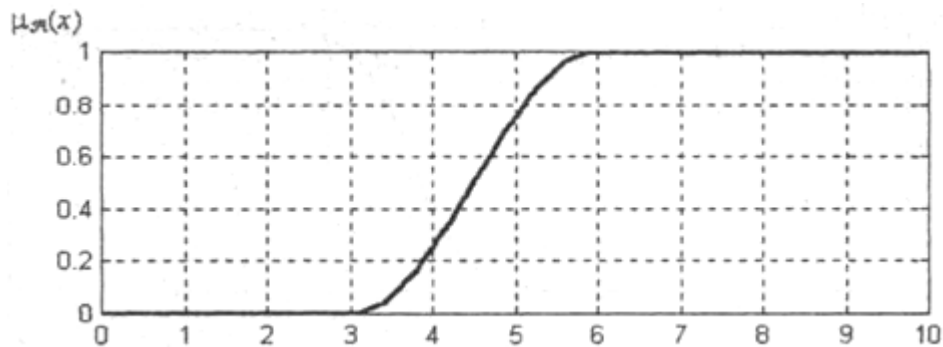
де a , b – деякі числові параметри, що набувають довільних дійсних значень й упорядковані відношенням $a < b$. Графік цієї функції для деякої нечіткої множини $\mathcal{A}$ і універсума $X=[0, 10]$ зображений на рис. 4.16,а, у цьому разі значення параметрів відповідно дорівнюють $a=3$, $b=6$.

Сплайн-функція може бути також задана виразом

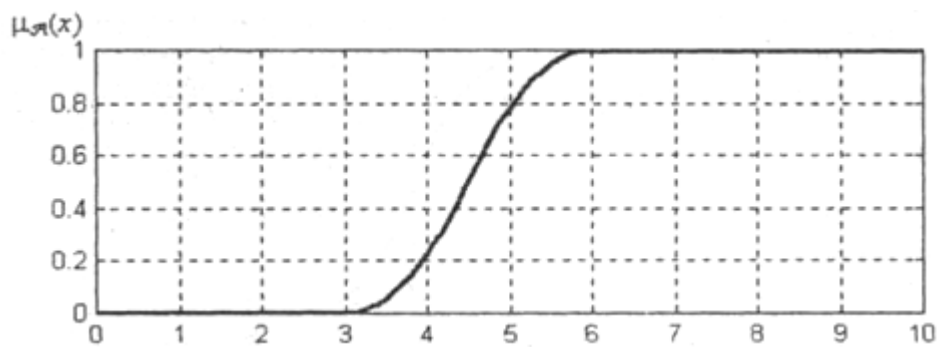
$$\mu_{S_2}(x; a, b) = \begin{cases} 0, & x \leq a \\ 2 \left(\frac{x-a}{b-a}\right)^2, & a < x \leq \frac{a+b}{2} \\ 1 - 2 \left(\frac{b-x}{b-a}\right)^2, & \frac{a+b}{2} < x < b \\ 1, & b \leq x \end{cases},$$

де a, b – деякі числові параметри, що набувають довільних дійсних значень й упорядковані відношенням $a < b$. Графік цієї функції для деякої нечіткої множини $\mathcal{A}$ і універсума $X=[0, 10]$ зображений на рис. 4.16,б, у цьому разі значення параметрів відповідно дорівнюють $a=3, b=6$.

Ці функції належності представляють нормальні випуклі нечіткі множини з ядром $[b, +\infty)$ і носієм $(a, +\infty)$.



a



б

Рис. 4.16. Графіки S-подібних функцій належності для значень параметрів $a=3, b=6$: *a* – fs_1 ; *б* – fs_2

Розглянуті S-подібні функції використовуються для наведення таких нечітких множин, які характеризуються невизначеністю типу: «велика кількість», «велике значення», «значна величина», «високий рівень доходів і цін», «висока напруга», «висока якість послуг», «великий струм» і багатьох інших. Загальним для всіх таких ситуацій є високий ступінь прояву тієї або іншої якісної або кількісної ознаки. Особливість нечіткого моделювання у цьому разі полягає в наведенні відповідних нечітких множин за допомогою монотонно наростаючих функцій належності.

У якості окремих випадків Z- і S-подібних кривих зручно розглядати так звану лінійну Z-подібну функцію (рис. 4.17,а) і лінійну S-подібну функцію (рис. 4.17,б).

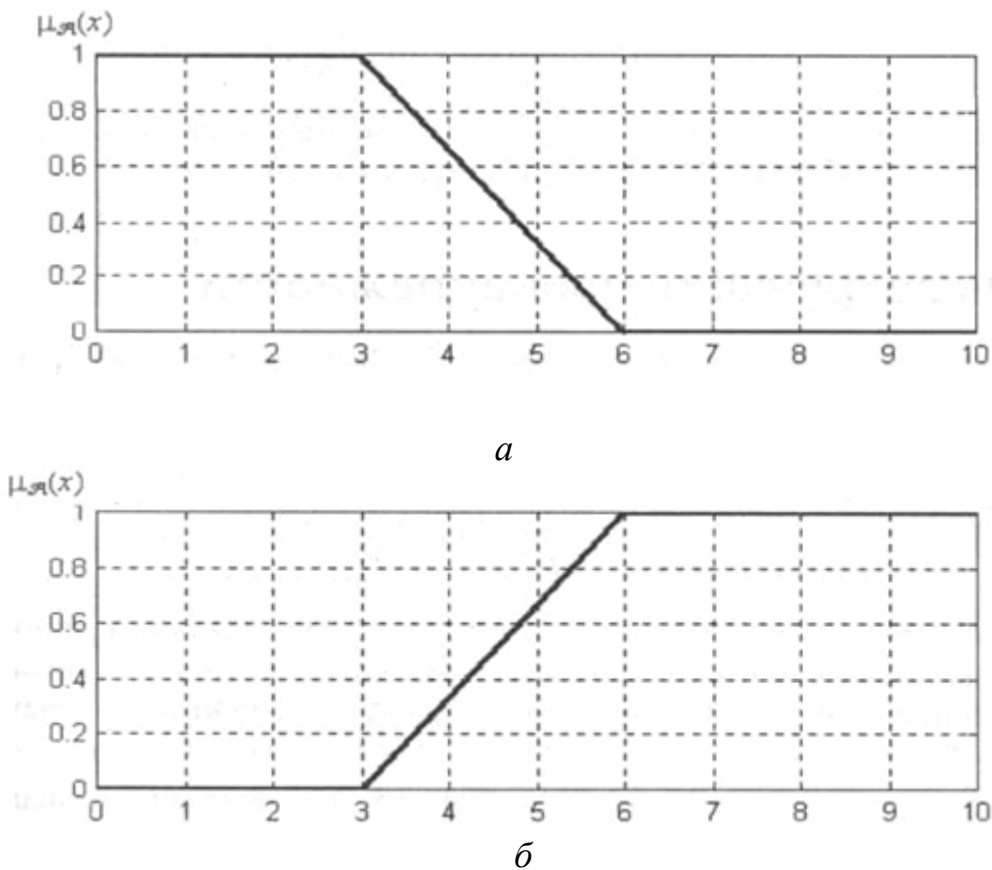


Рис. 4.17. Графіки функції належності для значень параметрів $a=3$, $b=6$: a – лінійної Z-подібної; $б$ – лінійної S-подібної

Перша з цих функцій у загальному випадку може бути задана аналітично таким виразом:

$$\mu_{\downarrow}(x; a, b) = \begin{cases} 1, & x \leq a \\ \frac{b-x}{b-a}, & a < x < b \\ 0, & b \leq x \end{cases},$$

де a , b – деякі числові параметри, що набувають довільних дійсних значень і упорядковані відношенням $a < b$. Графік цієї функції для деякої нечіткої множини $\mathcal{A}$ і універсума $\mathbf{X}=[0, 10]$ зображено на рис. 4.17, a , у цьому разі значення параметрів відповідно дорівнюють $a=3$, $b=6$.

Друга із цих функцій у загальному випадку може бути задана аналітично таким виразом:

$$\mu_{\uparrow}(x; a, b) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a < x < b \\ 1, & b \leq x \end{cases},$$

де a, b – деякі числові параметри, що набувають довільних дійсних значень й упорядковані відношенням $a < b$. Графік цієї функції для деякої нечіткої множини $\mathcal{A}$ і універсума $\mathbf{X}=[0, 10]$ зображено на рис. 4.17,б, у цьому разі значення параметрів відповідно дорівнюють $a=3, b=6$.

Ці функції належності представляють нормальні випуклі нечіткі множини із межами (a, b) .

Окрім розглянутих кусково–лінійних функцій належності часто використовуються гладкі монотонні нелінійно наростаючі та спадаючі форми функцій, такі як: гаусова, подвійна гаусова, узагальнена дзвоноподібна, сигмоподібна, подвійна сигмоподібна – різниця двох сигмоподібних *dsigmf*, подвійна сигмоподібна – добуток двох сигмоподібних *psigmf*, π –подібна (рис. 4.18 – рис. 4.24):

$$f(x; \sigma, c) = e^{\frac{-(x-c)^2}{2\sigma^2}};$$

$$\text{Сигмоїда } \sigma = \frac{1}{1 + e^{-x}};$$

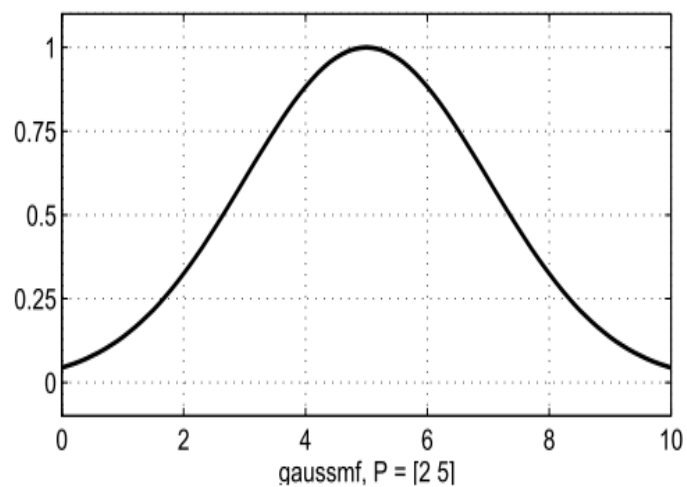


Рис. 4.18. Гаусова функція належності

$$f(x; \sigma, c) = (f(x_1; \sigma_1, c_1), f(x_2; \sigma_2, c_2));$$

$$f(x_1; \sigma_1, c_1) = e^{\frac{-(x_1-c_1)^2}{2\sigma_1^2}};$$

$$f(x_2; \sigma_2, c_2) = e^{\frac{-(x_2-c_2)^2}{2\sigma_2^2}};$$

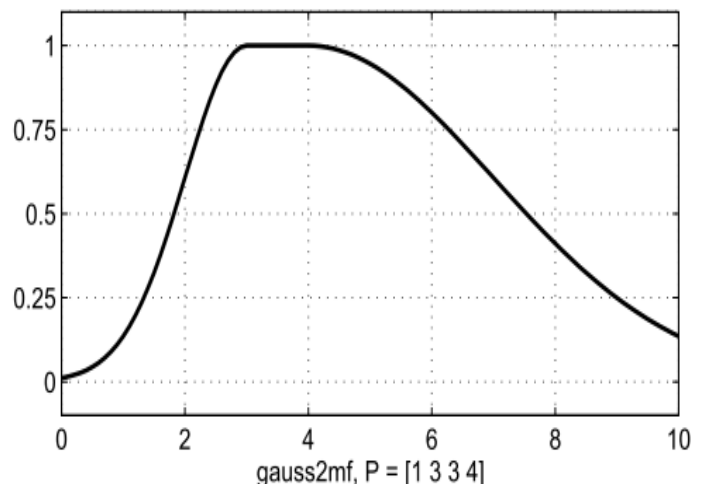


Рис.4.19. Подвійна гаусова функція належності

$$f(x;a,b,c) = \frac{1}{1 + \left| \frac{x-c}{a} \right|^{2b}} ;$$

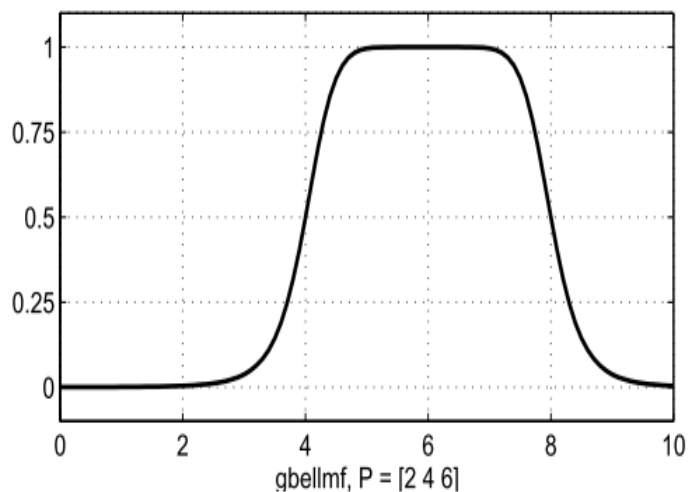


Рис. 4.20. Узагальнена дзвіноподібна функція

$$f(x,a,c) = \frac{1}{1 + e^{-a(x-c)}} ;$$

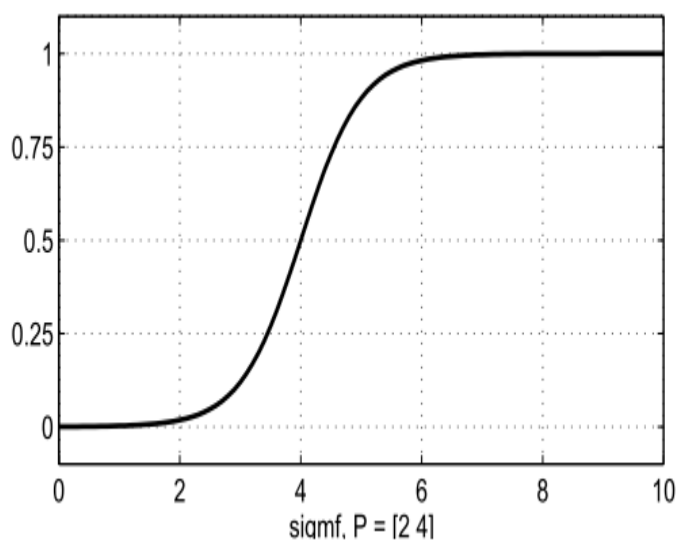


Рис. 4.21. Сигмоподібна функція

$$f(x;a,c) = f(x_1;a_1,c_1) - f(x_2;a_2,c_2) ;$$

$$f(x_1;a_1,c_1) = \frac{1}{1 + e^{-a_1(x_1-c_1)}} ;$$

$$f(x_2;a_2,c_2) = \frac{1}{1 + e^{-a_2(x_2-c_2)}} ;$$

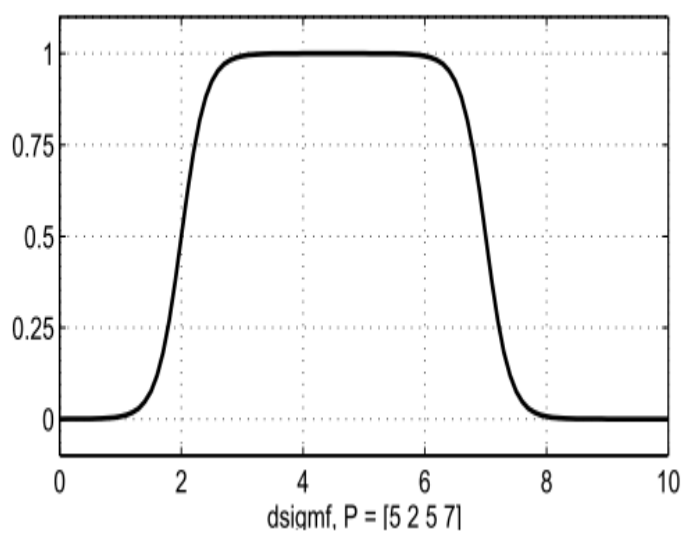


Рис. 4.22. Подвійна сигмоподібна функція різниця двох сигмоподібних

$$f(x; a, c) = f(x_1; a_1, c_1) \cdot f(x_2; a_2, c_2);$$

$$f(x_1; a_1, c_1) = \frac{1}{1 + e^{-a_1(x_1 - c_1)}};$$

$$f(x_2; a_2, c_2) = \frac{1}{1 + e^{-a_2(x_2 - c_2)}};$$

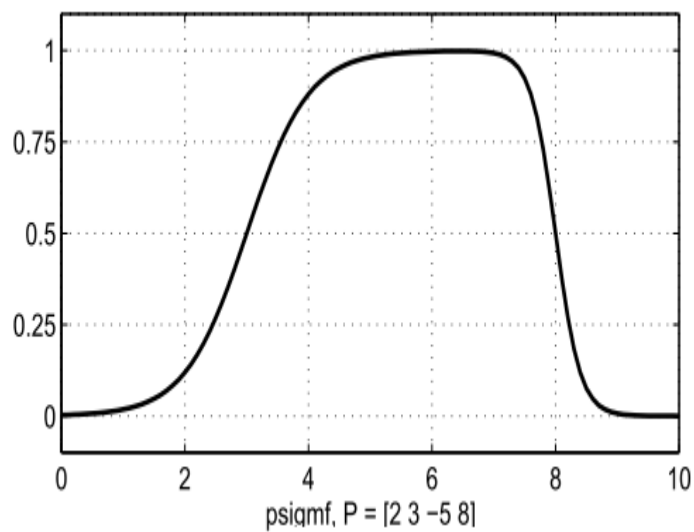


Рис.4.23. Подвійна сигмоподібна функція – добуток двох сигмоподібних

$$y = pimf(x, [a \ b \ c \ d]).$$

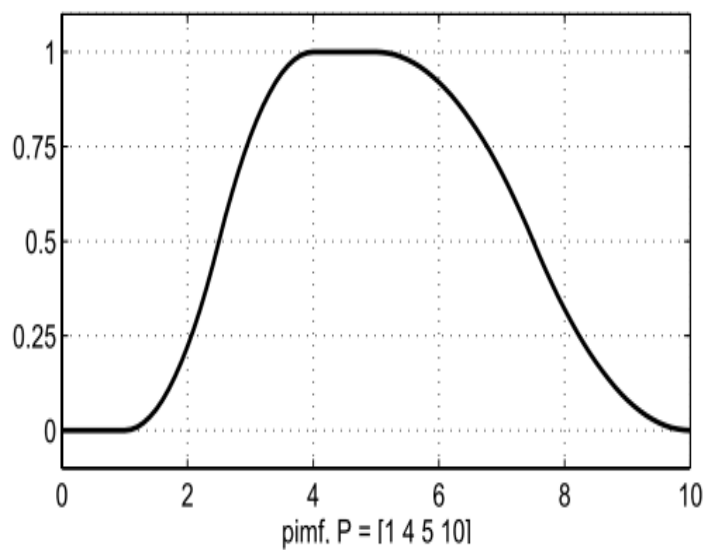


Рис. 4.24. π –подібна функція

4.7 Операції перетину, об'єднання і вкладення нечітких множин

Над нечіткими множинами можна виконувати різноманітні математичні операції за аналогією з класичними (чіткими). До найбільш поширених операцій над нечіткими множинами належать операції: визначення відношення вкладення, рівності, додаткової нечіткої множини, добутку (перетину) нечітких множин та суми (об'єднання) нечітких множин. Ці операції відповідно записуються в такому вигляді:

$$\begin{aligned}
 A \subset B &\leftrightarrow \mu_A(x) \leq \mu_B(x) && \text{для } \forall x \in X, \\
 A = B &\leftrightarrow \mu_A(x) = \mu_B(x) && \text{для } \forall x \in X, \\
 B = \bar{A}, A = \bar{B} &\leftrightarrow \mu_{\bar{A}}(x) = 1 - \mu_A(x) && \text{для } \forall x \in X, \\
 A \cap B &\leftrightarrow \mu_{A \cap B}(x) = \mu_A(x) \wedge \mu_B(x) = \min(\mu_A(x), \mu_B(x)) && \text{для } \forall x \in X, \\
 A \cup B &\leftrightarrow \mu_{A \cup B}(x) = \mu_A(x) \vee \mu_B(x) = \max(\mu_A(x), \mu_B(x)) && \text{для } \forall x \in X.
 \end{aligned}$$

Графічний алгоритм дії цих операцій продемонстровано на рис. 4.25.

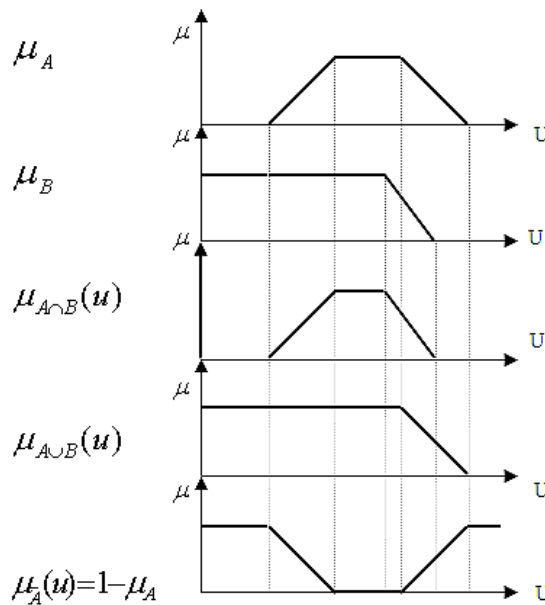


Рис. 4.25. Графічне зображення дії операцій $\mu_{A \cap B}(u)$, $\mu_{A \cup B}(u)$, $\mu_{\bar{A}}(u)$

Якщо позначити через $A_n(u)$ сукупність усіх нечітких множин у просторі діапазону U , то система $\{A_n(u), \overline{A_n(u)}, \subset, \cup, \cap\}$, очевидно, не утворюватиме булеву алгебру. Проте для операції $\subset$ справедливі рефлексивність, антисиметричність, транзитивність, а також розповсюджуються закони ідемпотентності, комутативності, асоціативності, подвійного заперечення і закон де Моргана. Але не виконується закон

компліментарності, оскільки мають місце співвідношення: $(A \cap \bar{A}) \supset 0$, $(A \cup \bar{A}) \subset U$. Тому вважають, що система $\{A_n(u), \overline{A_n(u)}, \subset, \cup, \cap\}$ утворює так звану повну псевдобулеву алгебру.

Крім зазначених вище операцій над нечіткими множинами важливими з практичної точки зору використання в системах нечіткого керування є такі операції, як: алгебраїчний добуток $A \times B$, граничний добуток $A \otimes B$ та драстичний (радикальний) добуток $A \Delta B$. Ці операції можна записати, відповідно, таким чином:

$$\mu_{A \times B}(u) = \mu_A(u) \times \mu_B(u) \quad \text{для } \forall u \in U,$$

$$\mu_{A \otimes B}(u) = (\mu_A(u) + \mu_B(u) - 1) \vee 0 \quad \text{для } \forall u \in U,$$

$$\mu_{A \Delta B}(u) = \begin{cases} \mu_B(u), & \text{якщо } \mu_A(u) = 1, \\ \mu_A(u), & \text{якщо } \mu_B(u) = 1 \text{ для } \forall u \in U, \\ 0 & \text{в інших випадках.} \end{cases}$$

На основі операції алгебраїчного добутку визначається операція піднесення до степеня α нечіткої множини, наприклад A , де α – додатне число. Нечітка множина A^α визначається функцією належності $\mu_{A^\alpha}(u) = \mu_A^\alpha(u)$. Окремими випадками піднесення до степеня є: операція концентрування A^2 (ущільнення) та операція розтягування $A^{0,5}$.

Використовуються також операції алгебраїчної суми $A \dot{+} B$, граничної суми $A \oplus B$, драстичної суми $A \nabla B$ та λ – суми $A \overset{+}{\lambda} B$ (λ – параметр, більший 0 та менший 1):

$$\mu_{A \dot{+} B}(u) = \mu_A(u) + \mu_B(u) - \mu_A(u)\mu_B(u) \quad \text{для } \forall u \in U,$$

$$\mu_{A \oplus B}(u) = (\mu_A(u) + \mu_B(u)) \wedge 1 \quad \text{для } \forall u \in U,$$

$$\mu_{A \nabla B}(u) = \begin{cases} \mu_B(u), & \text{якщо } \mu_A(u) = 0, \\ \mu_A(u), & \text{якщо } \mu_B(u) = 0 \text{ для } \forall u \in U, \\ 1 & \text{в інших випадках,} \end{cases}$$

$$\mu_{A \overset{+}{\lambda} B}(u) = \lambda \mu_A(u) + (1 - \lambda) \mu_B(u) \quad \text{для } \forall u \in U.$$

Крім того, використовуються і поняття різниці нечітких множин $A - B$ та абсолютної різниці $|A - B|$:

$$\mu_{A-B}(u) = (\mu_A(u) - \mu_B(u)) \vee 0 = \min(\mu_A(u), 1 - \mu_B(u)) \quad \text{для } \forall u \in U,$$

$$\mu_{|A-B|}(u) = |\mu_A(u) - \mu_B(u)| \quad \text{для } \forall u \in U.$$

Між добутком, сумою та λ -сумою нечітких множин має місце співвідношення вкладення:

$$0 \subset A \Delta B \subset A \otimes B \subset A \times B \subset A \cap B \subset A \overset{+}{\lambda} B \subset A \cup B \subset A \dot{+} B \subset \\ \subset A \oplus B \subset A \nabla B \subset X.$$

Основні операції з нечіткими множинами в розгорнутому вигляді наведемо на такому прикладі.

Приклад 4.1

Розглянемо дві множини A та B , при чому $A \neq B$ і $A \subset B$ на універсумі $U = u_1 + u_2 + \dots + u_4$:

$$A = 0,4/u_1 + 0,2/u_2 + 0/u_3 + 1/u_4;$$

$$B = 0,8/u_1 + 0,9/u_2 + 0,1/u_3 + 1/u_4.$$

Для цих множин можна записати:

$$\bar{A} = 0,6/u_1 + 0,8/u_2 + 1/u_3 + 0/u_4;$$

$$\bar{B} = 0,2/u_1 + 0,1/u_2 + 0,9/u_3 + 0/u_4;$$

$$A \cap B = 0,4/u_1 + 0,2/u_2 + 0/u_3 + 1/u_4;$$

$$A \cup B = 0,8/u_1 + 0,9/u_2 + 0,1/u_3 + 1/u_4;$$

$$A - B = A \cap \bar{B} = 0,2/u_1 + 0,1/u_2 + 0/u_3 + 0/u_4;$$

$$B - A = \bar{A} \cap B = 0,6/u_1 + 0,8/u_2 + 0,1/u_3 + 0/u_4.$$

Окремими випадками операції піднесення до степеня є операція концентрування, яка позначається таким чином:

$$\text{CON}(A) = A^2,$$

і операція розтягування

$$\text{DIL}(A) = A^{0.5}.$$

Приклад 4.2

Розглянемо дві множини A та B , при чому $A \neq B$ і $A \subset B$, на універсумі $U = 1 + 2 + \dots + 10$:

$$A = 0,8/3 + 1/5 + 0,6/6,$$

$$B = 0,7/3 + 1/4 + 0,5/6.$$

Для цих множин можна записати:

$$\bar{A} = 1/1+1/2+0,2/3+1/4+0,4/6+1/7+1/8+1/9+1/10,$$

$$A+B = 0,8/3+1/4+1/5+0,6/6,$$

$$A \cap B = 0,7/3+0,5/6,$$

$$AB = 0,56/3+0,3/6,$$

$$A^2 = 0,64/3+1/5+0,36/6,$$

$$0.4A = 0,32/3+0,4/5+0,24/6,$$

$$\text{CON}(B) = 0,49/3+1/4+0,25/6,$$

$$\text{DIL}(B) = 0,84/3+1/4+0,7/6.$$

В теорії нечітких множин розробляються питання побудови узагальнених параметризованих операторів перетину, об'єднання та доповнення для врахування різнобічних змістових відтінків відповідних їм комбінацій елементів «І», «АБО» та «НІ».

Нечіткі множини, основні поняття котрих розглянуті нами, називають множинами першого порядку. Окрім них мають місце і розширені поняття нечіткої множини, але з практичної точки зору в системах управління, зокрема в електротехніці, як правило, використовуються нечіткі множини першого порядку.

4.8 Математичні визначення нечіткої і лінгвістичної змінних

Нечітка змінна визначається як кортеж: $\langle \alpha, X, \mathcal{A} \rangle$, де α – найменування або назва нечіткої змінної; X – область її визначення (універсум); $\mathcal{A} = \{x, \mu_{\mathcal{A}}(x)\}$ – нечітка множина на універсумі X , що описує можливі значення, яких може набувати нечітка змінна α . Таким чином, говорячи про нечітку змінну α , ми завжди будемо мати на увазі деяку нечітку множину $\mathcal{A}$, яка визначає її можливі значення.

Як нечітку змінну можна представити нечітку множину, наприклад, $\mathcal{B}$, яка характеризує підвищені значення змінної напруги мережі живлення – «висока напруга». У цьому випадку відповідна нечітка змінна може бути представлена в такий спосіб:

$$\langle \text{висока напруга}, \{u | 110 \text{ кВ} < u < 121 \text{ кВ}\}, \mathcal{B} \rangle,$$

де $\mathcal{B} = \{u, \mu_{\mathcal{B}}(u)\}$ – нечітка множина із функцією належності $\mu_{\mathcal{B}}(u)$.

Лінгвістична змінна також визначається як кортеж: $\langle \beta, T, X, G, M \rangle$, де β – найменування або назва лінгвістичної змінної; T – базова терм-множина лінгвістичної змінної або множина її значень (*термів*), кожне з яких є найменуванням окремої нечіткої змінної α ;

X – область визначення (універсум) нечітких змінних, які входять у визначення лінгвістичної змінної β ;

G – деяка синтаксична процедура, яка описує процес утворення чи генерування з множини T нових, осмислених у розглянутому контексті значень для такої лінгвістичної змінної;

M – семантична процедура, яка дає змогу поставити у відповідність кожному новому значенню цієї лінгвістичної змінної, отриманому за допомогою процедури G , деякий осмислений зміст за допомогою формування відповідної нечіткої множини.

Наприклад, формалізація суб'єктивної оцінки допустимих вхідних значень змінної напруги стабілізатора напруги 220 В мережі живлення може бути виконана за допомогою такої лінгвістичної змінної: $\langle \beta_1, T, X, G, M \rangle$, де β_1 – напруга мережі;

$T = \{ \langle \text{«низька»}, \text{«нормальна»}, \text{«висока»} \rangle \}$;

$X = [90 \text{ В}, 242 \text{ В}]$;

G – процедура створення нових термів за допомогою сполучень логічних сполучень «І», «АБО» і модифікаторів типу «дуже», «НІ», «злегка» та ін. («дуже низька», «дуже висока»);

M – процедура введення на $X=[90, 242]$ нечітких змінних $\alpha_1=\text{«низька»}$, $\alpha_2=\text{«нормальна»}$, $\alpha_3=\text{«висока»}$, а також відповідних нечітких множин для термів з $G(T)$ відповідно до правил трансляції нечітких сполучень і модифікаторів «І», «АБО», «НІ», «дуже», «злегка».

Для розглянутого прикладу нечіткі множини A_1, A_2, A_3 , що відповідають нечітким змінним: $\alpha_1 = \text{«низька»}$, $\alpha_2 = \text{«нормальна»}$, $\alpha_3 = \text{«висока»}$, для зручності задають графічно за допомогою кусково-лінійних функцій належності. Графіки функцій належності декількох нечітких змінних, які використовуються для представлення однієї лінгвістичної змінної, зображують на одному рисунку, що дає змогу порівнювати значення функцій належності відповідних нечітких змінних для різних значень універсуму.

4.8.1 Поняття нечіткого висловлювання

У запропонованому Л. Заде варіанті нечіткої логіки множина істинних значень висловлювань узагальнюється до інтервалу дійсних значень $[0,1]$, що дає змогу висловлюванню ухвалювати будь-яке значення істинності із цього інтервалу. Це чисельне значення є кількісною оцінкою ступеня істинності висловлювання, щодо якого не можна з повною впевненістю сказати про його істинність або хибність. Використання в якості множини істинних значень інтервалу $[0, 1]$ дає змогу побудувати логічну систему, у рамках якої виявилось можливим виконувати міркування з невизначеністю та оцінювати істинність висловлювань типу: *«Напруга в мережі досить висока»*, *«Струм в колі досить значний»*, *«Потужність навантаження гранично низька»* та ін.

Важливим (системоутворюючим) поняттям нечіткої логіки є поняття елементарного нечіткого висловлювання.

У загальному випадку елементарним нечітким висловлюванням називається *оповідальна пропозиція, що виражає закінчену думку, щодо якої ми можемо судити про її істинність або хибність тільки з деяким ступенем впевненості.*

Елементарні нечіткі висловлювання для зручності позначаються тими ж літерами, що й нечіткі множини: $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$, $\mathcal{D}$, $\mathcal{E}$. Самі елементарні нечіткі вирази іноді називають просто нечіткими виразами.

Головною відмінністю елементарного нечіткого висловлювання від елементарного висловлювання математичної логіки є наступний факт. Множина значень істинності елементарних висловлювань класичної математичної логіки складається із двох елементів: {«*правда*», «*неправда*»} ({П, Н} або {0, 1}), у цьому разі значенню «*правда*» відповідає цифра 1 або літера П, а значенню «*неправда*» – цифра 0 або літера Н.

У нечіткій логіці ступінь істинності елементарного нечіткого висловлювання набуває значення із замкненого інтервалу [0, 1], причому 0 і 1 є граничними значеннями ступеня істинності і збігаються зі значеннями «*неправда*» і «*правда*» відповідно.

Нечітким лінгвістичним висловлюванням будемо називати висловлювання таких трьох видів:

1. Висловлювання « $\beta \in \alpha$ », де β – найменування лінгвістичної змінної, α – її значення, якому відповідає окремий лінгвістичний терм (нечітка змінна) із базової терм-множини T лінгвістичної змінної β .
2. Висловлювання « $\beta \in \nabla \alpha$ », де ∇ – модифікатор, відповідає таким словам, як: «*ДУЖЕ*», «*БІЛЬШ-МЕНШ*», «*НАБАГАТО БІЛЬШЕ*» та іншим, які можуть бути отримані з використанням певних процедур над такою лінгвістичною змінною.
3. Складні висловлювання, утворені з висловлювань видів 1 і 2 і нечітких логічних операцій у формі сполучень: «*І*», «*АБО*», «*ЯКБИ-ТО*», «*НІ*».

Оскільки в системах нечіткого висновку нечіткі лінгвістичні висловлювання займають центральне місце, далі будемо їх називати просто нечіткими висловлюваннями.

Наприклад, розглянемо деякі приклади нечітких висловлювань. Перше з них – «*напруга мережі є висока*» є нечітким висловлюванням першого виду, у рамках якого лінгвістичній змінній «*напруга побутової мережі*» присвоюється значення «*висока*». У цьому разі передбачається, що на універсальній множині U змінної «*напруга мережі*» визначений відповідний лінгвістичний терм «*висока*», який задається у формі трикутної функції належності деякої нечіткої множини (рис. 4. 26, а).

Нечітке висловлювання другого виду «*напруга мережі є дуже висока*» означає, що лінгвістичній змінній «*напруга мережі*» присвоюється значення «*висока*» з модифікатором «*ДУЖЕ*», який змінює значення відповідного лінгвістичного терма «*висока*» на основі використання деякої розрахункової формули, наприклад піднесення до степеня (більшого одиниці) нечіткої множини терма «*висока*».

Нечітке висловлювання другого виду «*напруга мережі є більш–менш висока*» означає, що лінгвістичній змінній «*напруга мережі*» присвоюється значення «*висока*» з модифікатором «*БІЛЬШ–МЕНШ*», який змінює значення відповідного лінгвістичного терма «*висока*» на основі використання деякої розрахункової формули, наприклад операції розтягування, тобто піднесення до степеня (меншого одиниці) нечіткої множини $\mathcal{A}$ для терма «*висока*».

На рис. 4.26 представлені функції належності терм–множини «*висока*» лінгвістичної змінної «*напруга побутової мережі*» (рис. 4.26,а) і визначення значень функцій належності цієї ж терм–множини «*висока*» з модифікаторами «*ДУЖЕ*» (рис. 4.26,б) і «*БІЛЬШ МЕНШ*» (рис. 4.26,в).

Розглянемо нечітке висловлювання третього виду «*напруга мережі є висока і неузгодження з номінальним струмом є значне*» означає, що одній лінгвістичній змінній «*напруга мережі*» присвоюється значення «*висока*», а іншій лінгвістичній змінній «*неузгодження з номінальним струмом*» присвоюється значення «*значне*». Ці нечіткі висловлювання першого виду з'єднані логічною операцією нечітка кон'юнкція (операцією нечіткого «І»).

4.8.2 Основні логічні операції з нечіткими висловлюваннями

Нехай U – деяка множина елементарних нечітких висловлювань, а $T: U \rightarrow [0, 1]$ – відображення істинності висловлювань.

Логічне заперечення. Запереченням нечіткого висловлювання $\mathcal{A}$ (записується як $\neg \mathcal{A}$ і читається як «не $\mathcal{A}$ », «невірно, що $\mathcal{A}$ ») називається унарна логічна операція, результат якої є нечітким висловлюванням, істинність якого за визначенням набуває значення

$$T(\neg \mathcal{A}) = 1 - T(\mathcal{A}).$$

Очевидно, що прийнятий у математичній логіці спосіб визначати логічні операції за допомогою таблиць істинності не може бути використаний у нечіткій логіці. Причина цього полягає в континуальній потужності множини істинних значень $[0, 1]$. Саме тому центральну роль у визначенні логічних операцій у нечіткій логіці здобуває відображення істинності T , яке має допоміжне значення в класичній математичній логіці.

Прикладом застосування операції логічного заперечення до нечіткого висловлювання «*На виході буде низька напруга*» є висловлювання «*На виході*

буде не низька напруга», ступінь істинності якого, якщо прийняти $T(\mathcal{A}) = 0,2$, набуває значення 0,8.

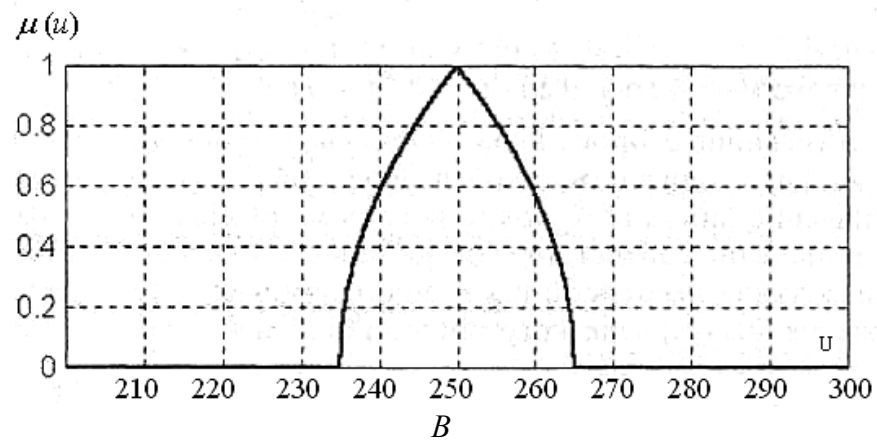
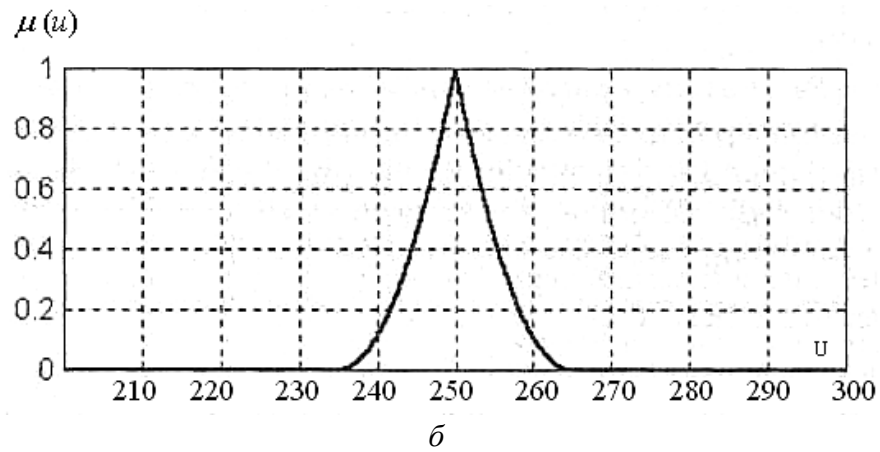
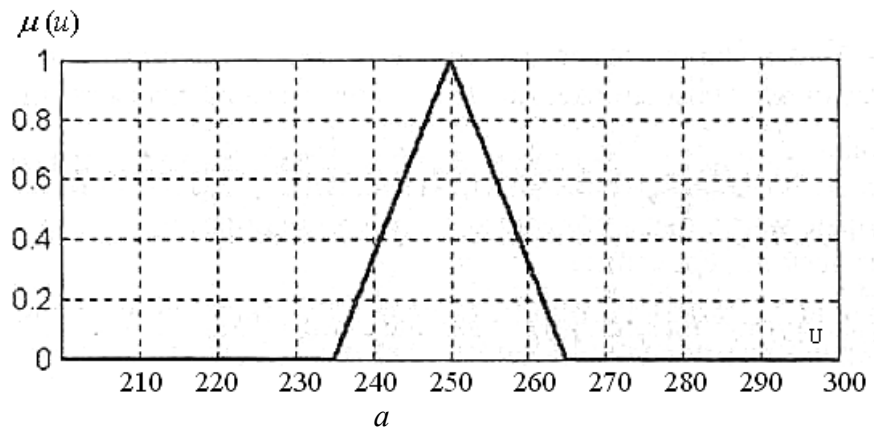


Рис. 4.26. Застосування модифікаторів «ДУЖЕ» (б) і «БІЛЬШ МЕНШ» (в) до терм-множини «висока» (а) лінгвістичної змінної U «напруга побутової мережі»

Логічна кон'юнкція. Кон'юнкцією нечітких висловлювань $\mathcal{A}$ і $\mathcal{B}$ (записується як $\mathcal{A} \wedge \mathcal{B}$ і читається як « $\mathcal{A}$ і $\mathcal{B}$ ») називається бінарна логічна операція, результат якої є нечітким висловлюванням, істинність якого визначається за формулою

$$T(\mathcal{A} \wedge \mathcal{B}) = \min\{T(\mathcal{A}), T(\mathcal{B})\}.$$

Логічну кон'юнкцію нечітких висловлювань також називають *нечітким логічним «І»*, *нечіткою кон'юнкцією* або *min-кон'юнкцією* та іноді записують також у формі $\mathcal{A}$ AND $\mathcal{B}$. У цьому разі прийнято вважати наведену формулу *основною* для визначення ступеня істинності її результату.

Логічна диз'юнкція. Диз'юнкцією нечітких висловлювань $\mathcal{A}$ і $\mathcal{B}$ (записується як $\mathcal{A} \vee \mathcal{B}$ і читається як « $\mathcal{A}$ або $\mathcal{B}$ ») називається бінарна логічна операція, результат якої є нечітким висловлюванням, істинність якого за визначенням набуває значення

$$T(\mathcal{A} \vee \mathcal{B}) = \max\{T(\mathcal{A}), T(\mathcal{B})\}.$$

Логічну диз'юнкцію нечітких висловлювань також називають *нечітким логічним невиключним «АБО»*, *нечіткою диз'юнкцією* чи *max-диз'юнкцією* та інколи записують також у формі $\mathcal{A}$ OR $\mathcal{B}$. У цьому разі прийнято вважати наведену формулу *основною* для визначення ступеня істинності її результату.

Нечітка імплікація. Нечіткою імплікацією або просто *імплікацією* нечітких висловлювань $\mathcal{A}$ і $\mathcal{B}$ (записується як $\mathcal{A} \supset \mathcal{B}$ і читається як «з $\mathcal{A}$ випливає $\mathcal{B}$ », «ЯКЩО $\mathcal{A}$, ТО $\mathcal{B}$ ») називається бінарна логічна операція, результат якої є нечітким висловлюванням, істинність якого може набувати значення, обумовленого за однією з таких формул.

– Класична нечітка імплікація, запропонована Л. Заде:

$$T(\mathcal{A} \supset \mathcal{B}) = \max\{\min\{T(\mathcal{A}), T(\mathcal{B})\}, 1 - T(\mathcal{A})\}.$$

Цю форму нечіткої імплікації називають також *нечіткою імплікацією Заде*.

– Класична нечітка імплікація для випадку $T(\mathcal{A}) \geq T(\mathcal{B})$:

$$T(\mathcal{A} \supset \mathcal{B}) = \max\{T(\neg \mathcal{A}), T(\mathcal{B})\} = \max\{1 - T(\mathcal{A}), T(\mathcal{B})\}.$$

Цю форму нечіткої імплікації іноді називають *нечіткою імплікацією Гьоделя*.

– Нечітка імплікація, запропонована Е. Мамдані:

$$T(\mathcal{A} \supset \mathcal{B}) = \min\{T(\mathcal{A}), T(\mathcal{B})\}.$$

Цю форму нечіткої імплікації також називають *нечіткою імплікацією Мамдані*, або нечіткою імплікацією *мінімуму кореляції*. Можна помітити, що у випадку $T(\mathcal{A}) \geq 0,5$ і $T(\mathcal{B}) \geq 0,5$ класична нечітка імплікація перетворюється в нечітку імплікацію Мамдані.

Нечітка еквівалентність. *Еквівалентністю* нечітких висловлювань $\mathcal{A}$ і $\mathcal{B}$, або просто нечіткою еквівалентністю (записується як $\mathcal{A} \equiv \mathcal{B}$ і читається як « $\mathcal{A}$ еквівалентно $\mathcal{B}$ ») називається бінарна логічна операція, результат якої є нечітким висловлюванням, істинність якого визначається за такою формулою:

$$T(\mathcal{A} \equiv \mathcal{B}) = \min \{ \max \{ T(\neg \mathcal{A}), T(\mathcal{B}) \}, \max \{ T(\mathcal{A}), T(\neg \mathcal{B}) \} \}.$$

Прикладом логічної еквівалентності може служити складне нечітке висловлювання: «*Струм в лінії електропередавання є високим еквівалентно тому, що потужність навантаження значна*», істинність якого набуває значення 0,8.

Як у класичній математичній логіці, так і в нечіткій логіці за допомогою розглянутих логічних сполучень можуть бути утворені досить складні нечіткі висловлювання. У цьому разі для явної вказівки порядку їх формування використовуються круглі дужки, а іноді – і пріоритет відповідних нечітких логічних операцій.

Наприклад, складне висловлювання $(\mathcal{A}_1 \supset \mathcal{A}_2) \supset (\neg \mathcal{A}_2 \supset \neg \mathcal{A}_1)$ у випадку $T(\mathcal{A}_1) = 0,7$ і $T(\mathcal{A}_2) = 0,2$ має значення істинності, що дорівнює:

$$\begin{aligned} & \max \{ 1 - \max \{ 1 - T(\mathcal{A}_1), T(\mathcal{A}_2) \} \}, \max \{ 1 - T(\neg \mathcal{A}_2), T(\neg \mathcal{A}_1) \} \} = \\ & = \max \{ 1 - \max \{ 1 - 0,7; 0,2 \} \}, \max \{ 1 - 0,8; 0,3 \} \} = \max \{ 0,7; 0,3 \} = 0,7. \end{aligned}$$

Обов'язковою умовою коректності визначення істинності складних нечітких висловлювань є вимога одночасної підстановки замість однакових термів (літер) саме їх нечітких висловлювань. Крім розглянутих логічних операцій можуть бути визначені й інші бінарні логічні операції з нечіткими висловлюваннями. Більш детально познайомитися з ними можна в спеціальній літературі.

4.9 Правила нечітких знань – продукцій

Продукційні системи були розроблені в рамках досліджень за методами штучного інтелекту і знайшли широке застосування для наведення знань і висновків у експертних системах, заснованих на правилах. Оскільки нечіткий висновок реалізується на основі нечітких продукційних правил, розгляд базового формалізму нечітких моделей набуває самостійного значення. У

цьому разі нечіткі правила продукцій не тільки багато в чому близькі до логічних моделей, але і, що особливо важливо, дають змогу адекватно представити практичні знання експертів у тій чи іншій проблемній сфері.

Правило нечіткої продукції (бази знань). У загальному випадку під *правилом нечітких знань* або просто – нечіткою продукцією розуміють вираз такого виду:

$$(i) : Q; P; \mathcal{A} \Rightarrow \mathcal{B}; S, F, N,$$

де (i) – ім'я нечіткої продукції; Q – сфера застосування нечіткої продукції; P – умова застосування ядра нечіткої продукції; $\mathcal{A} \Rightarrow \mathcal{B}$ – ядро нечіткої продукції, в якому $\mathcal{A}$ – умова ядра (або антецедент); $\mathcal{B}$ – висновок ядра (або консеквент); « $\Rightarrow$ » – знак логічної секвенції (або походження); S – метод, або спосіб визначення кількісного значення ступеня істинності висновку ядра; F – коефіцієнт визначеності або упевненості нечіткої продукції (ваговий коефіцієнт); N – коментар до правила.

За аналогією зі звичайним правилом продукції у якості імені (i) нечіткої продукції може виступати та або інша сукупність літер чи символів, що дає змогу однозначно ідентифікувати нечітку продукцію в системі нечіткого виводу або базі нечітких правил. У якості імені нечіткої продукції може використовуватися її номер у базі.

Сфера застосування нечіткої продукції Q , умова застосування ядра нечіткого висновку продукції P і коментар нечіткої продукції N визначаються аналогічно звичайній (не нечіткій) класичній системі правил.

Аналогічно звичайним правилам продукцій ядро $\mathcal{A} \Rightarrow \mathcal{B}$ також є центральним компонентом нечіткої продукції. Ядро продукції записується в більш звичній формі як «ЯКЩО $\mathcal{A}$, ТО $\mathcal{B}$ » або в найпоширенішому виді «IF $\mathcal{A}$, THEN $\mathcal{B}$ », де $\mathcal{A}$ і $\mathcal{B}$ – деякі вирази нечіткої логіки, які найбільш часто наводяться у формі нечітких висловлювань. У цьому разі секвенція інтерпретується у звичайному логічному розумінні як знак логічного походження висновку $\mathcal{B}$ з умови $\mathcal{A}$. У якості виразів $\mathcal{A}$ і $\mathcal{B}$ можуть використовуватися складні логічні нечіткі висловлювання, тобто елементарні нечіткі висловлювання, з'єднані нечіткими логічними сполучниками, такими як нечітке заперечення, нечітка кон'юнкція і нечітка диз'юнкція.

S – метод або спосіб визначення кількісного значення ступеня істинності висновку $\mathcal{B}$ на основі відомого значення ступеня істинності умови $\mathcal{A}$. Цей спосіб у загальному випадку визначає так звану *схему* або алгоритм нечіткого висновку в продукційних нечітких системах і називається також *методом композиції*, або *методом активації* згідно зі стандартом ІЕС 1131–7. На цей час для такої мети запропоновано кілька способів, основні з яких розглядаються нижче в цьому розділі.

F – коефіцієнт визначеності або впевненості виражає кількісну оцінку ступеня істинності або відносну вагу правила (нечіткої продукції). Коефіцієнт впевненості набуває свого значення з інтервалу $[0,1]$ і часто називається ваговим коефіцієнтом нечіткого правила продукції.

Зауваження

Порядок розміщення правил у базі знань не має значення.

Продукційна нечітка система (база знань).
Продукційна нечітка система, або система нечітких правил є деякою множиною окремих нечітких правил у формі «ЯКЩО A , ТО B » (або у вигляді «IF A THEN B », як визначено в стандарті ІЕС 1131–7).

Основна особливість нечітких правил, що використовуються в системах нечіткого логічного висновку, – це те, що умови та висновки окремих нечітких правил формулюються у формі нечітких висловлювань виду 1, 2 або 3 (розділ 4.8.1) щодо значень тих або інших лінгвістичних змінних.

Найпростіший варіант правила нечіткої продукції, який найбільш часто використовується в системах нечіткого висновку, може бути записаний у формі

ПРАВИЛО $\langle \# \rangle$: ЯКЩО « $\beta_1 \in \alpha_1'$ », ТО « $\beta_2 \in \alpha_2'$ ».

Тут нечітке висловлювання « $\beta_1 \in \alpha_1'$ » є умовою цього правила, а нечітке висловлювання « $\beta_2 \in \alpha_2'$ » – нечітким висновком цього правила. У цьому разі вважається, що $\beta_1 \neq \beta_2$.

Основна проблема наближених міркувань із використанням нечітких правил полягає в тому, щоб на основі деяких нечітких висловлювань із відомим ступенем істинності, які є умовами нечітких правил, оцінити ступінь істинності інших нечітких висловлювань, що є висновками відповідних нечітких правил.

Щоб мати можливість розв'язати цю проблему, необхідно відповісти на більш конкретне питання:

Чому має дорівнювати ступінь істинності висновку окремого нечіткого правила продукції, якщо відомий ступінь істинності умови цього правила?

Таким чином, у системах нечітких баз знань або продукцій центральне місце займає спосіб чи метод визначення істинності висновків у нечіткому правилі.

Взаємозв'язок між умовою та висновком у нечіткому правилі в загальному випадку є деяким бінарним нечітким відношенням на декартовому добутку універсумів відповідних нечітких висловлювань.

У загальному випадку для формального визначення різних методів нечіткого висновку стосовно нечіткого правила розглянемо дві нечіткі

множини $\mathcal{A}$ і $\mathcal{B}$, задані відповідно на універсумах X і Y . У цьому разі нечітка множина $\mathcal{A}$ інтерпретується як умова деякого нечіткого правила продукції, а нечітка множина $\mathcal{B}$ – як висновок цього ж правила.

Основна ідея полягає в тому, що нечітка множина $\mathcal{A}$ може розглядати як унарне відношення на універсумі X , а нечітка множина $\mathcal{B}$ може розглядати як унарне відношення на універсумі Y . У цьому випадку перше відношення визначається функцією належності $\mu_{\mathcal{A}}(x)$, а друге – функцією належності $\mu_{\mathcal{B}}(y)$.

Припустимо, що деяким чином визначено бінарне нечітке відношення на декартовому добутку універсумів:

$$Q = \{ \langle x, y \rangle, \mu_Q(\langle x, y \rangle) \}, \text{ де } x \in X \text{ і } y \in Y.$$

Якщо додатково відома функція належності $\mu_{\mathcal{A}}(x)$ першої множини, то функція належності $\mu_{\mathcal{B}}(y)$ другої множини може бути визначена в результаті нечіткої композиції відповідних нечітких відносин з використанням, наприклад, формули для максимінної (*max–min*) нечіткої композиції.

Окрім максимінної нечіткої композиції для визначення функції належності нечіткої множини $\mathcal{B}$ використовуються й інші методи, засновані на різних розрахункових формулах:

- **Max–min** – композиція або максимінна нечітка згортка:

$$\mu_{\mathcal{B}}(y) = \max_{x \in X} \{ \min \{ \mu_{\mathcal{A}}(x), \mu_Q(\langle x, y \rangle) \} \}.$$

- **Max–prod** – композиція:

$$\mu_{\mathcal{B}}(y) = \max_{x \in X} \{ \mu_{\mathcal{A}}(x) \cdot \mu_Q(\langle x, y \rangle) \}.$$

- **Min–max** – композиція:

$$\mu_{\mathcal{B}}(y) = \min_{x \in X} \{ \max \{ \mu_{\mathcal{A}}(x), \mu_Q(\langle x, y \rangle) \} \}.$$

- **Max–max** – композиція:

$$\mu_{\mathcal{B}}(y) = \max_{x \in X} \{ \max \{ \mu_{\mathcal{A}}(x), \mu_Q(\langle x, y \rangle) \} \}.$$

– **Min–min**–композиція:

$$\mu_B(y) = \min_{x \in X} \{ \min \{ \mu_A(x), \mu_Q(<x, y>) \} \}.$$

– **Max–average** – композиція:

$$\mu_B(y) = 0.5 \max_{x \in X} \{ \mu_A(x) + \mu_Q(<x, y>) \}.$$

– **Sum–prod**–композиція:

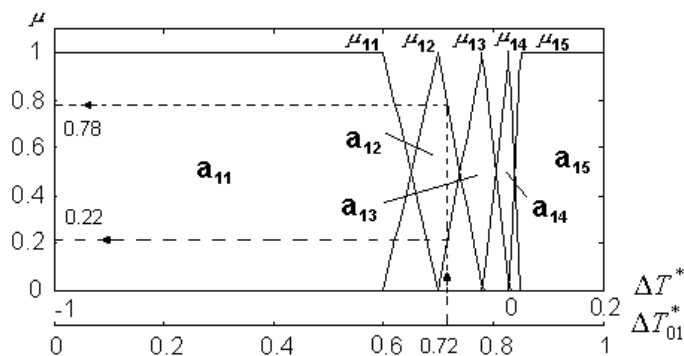
$$\mu_B(y) = f\left(\sum_{x \in X} (\mu_A(x) \cdot \mu_Q(<x, y>))\right),$$

де f – деяка логічна функція типу сигмоїдної, яка обмежує значення функції числом з інтервалу $[0, 1]$. Цей метод композиції застосовується в додатках штучних нейронних мереж для встановлення взаємозв'язків між паралельними шарами в багатошарових мережах.

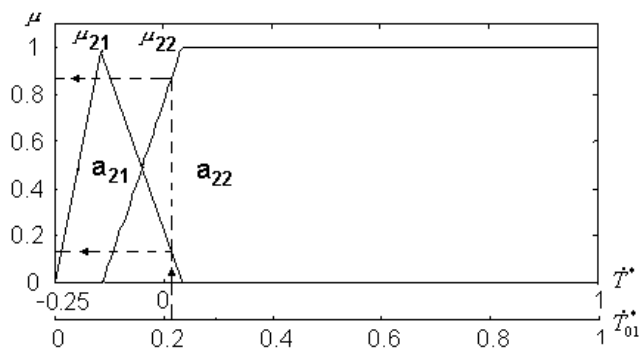
Приклад 4.3

Синтез НЛК для регулювання напруги в електропечі за критерієм заданої температури.

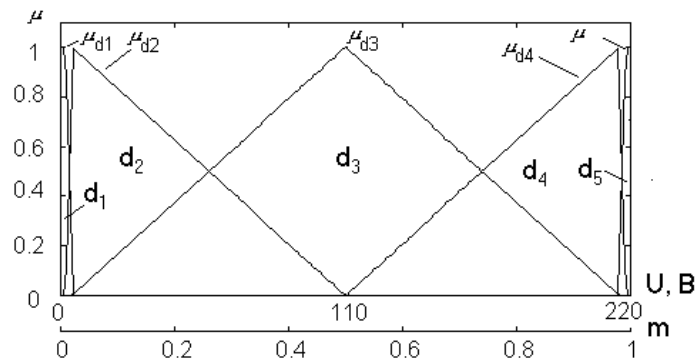
1. Представлення лінгвістичної змінної (вхідна – *приріст температури*) A_1



2. Представлення лінгвістичної змінної (вхідна – *температура*) A_2



3. Представлення лінгвістичної змінної (вихідна – коефіцієнт напруги або кількість періодів напруги в інтервалі регулювання) D



4. Створення бази знань НЛК

Назва правила	Синтаксис правила	Ваговий коефіцієнт
R1	<i>ЯКЩО</i> $A_1 \in a_{11}$ <i>І</i> $A_2 \in a_{22}$ <i>ТО</i> $D \in d_5$	1
R2	<i>ЯКЩО</i> $A_1 \in a_{12}$ <i>І</i> $A_2 \in a_{22}$ <i>ТО</i> $D \in d_4$	1
R3	<i>ЯКЩО</i> $A_1 \in a_{13}$ <i>І</i> $A_2 \in a_{22}$ <i>ТО</i> $D \in d_3$	1
R4	<i>ЯКЩО</i> $A_1 \in a_{14}$ <i>І</i> $A_2 \in a_{22}$ <i>ТО</i> $D \in d_2$	0,6
R5	<i>ЯКЩО</i> $A_1 \in a_{15}$ <i>І</i> $A_2 \in a_{22}$ <i>ТО</i> $D \in d_1$	1
R6	<i>ЯКЩО</i> $A_1 \in a_{15}$ <i>І</i> $A_2 \in a_{21}$ <i>ТО</i> $D \in d_1$	1

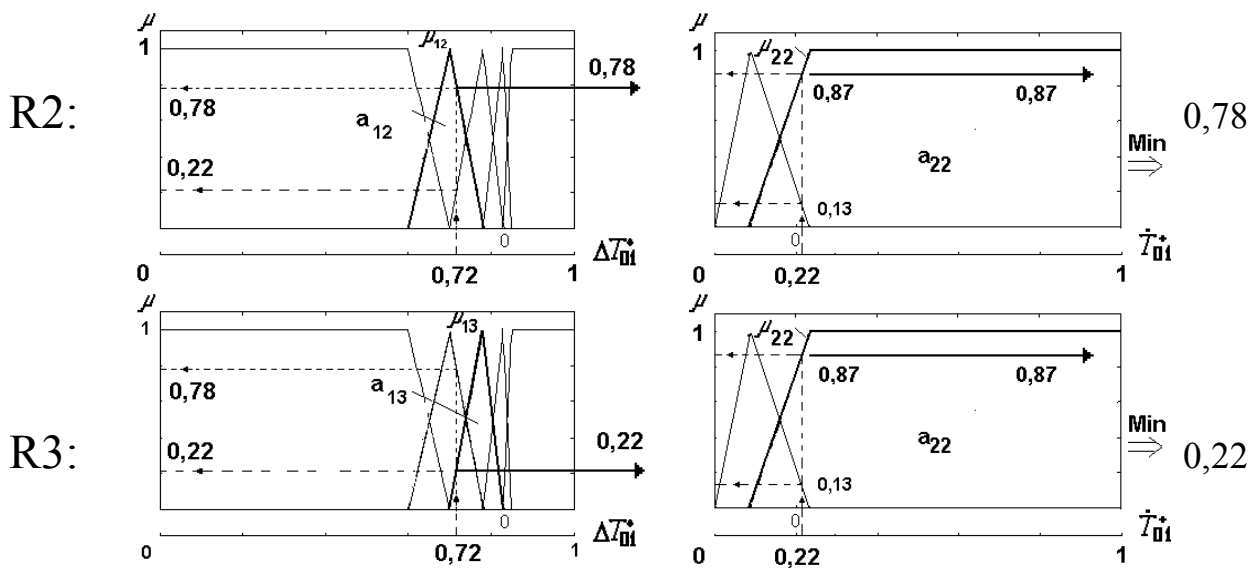
5. Процес об'єднання підумов (знаходження ступеня істинності умови) – агрегація:

$$\text{Правило R2: } \mu_{y,R2} = \text{Min}\{\mu_{12}, \mu_{22}\};$$

$$\text{Правило R3: } \mu_{y,R3} = \text{Min}\{\mu_{13}, \mu_{22}\}.$$

$$\text{Тобто } \mu_{y,R2} = \text{Min}\{0,78; 0,87\} = 0,78;$$

$$\mu_{y,R3} = \text{Min}\{0,22; 0,87\} = 0,22.$$

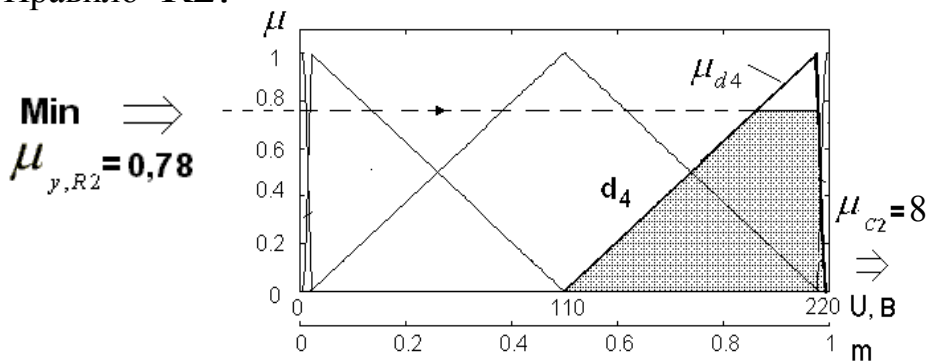


6. Процес активації або композиції (імплікація)

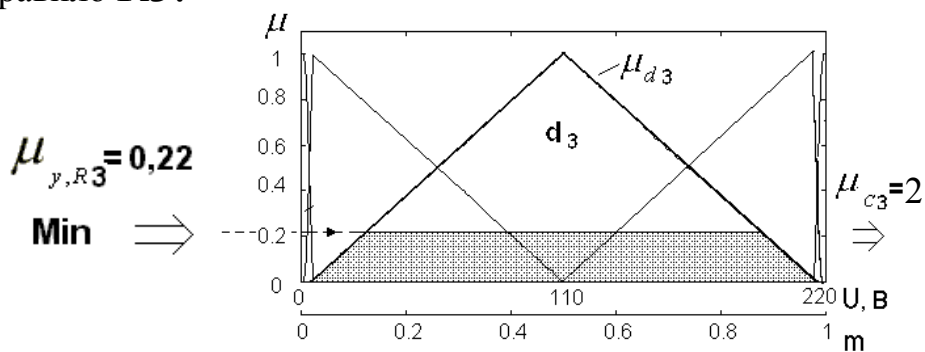
Правило R2: $\mu_{c2} = \text{Min}\{\mu_{y.R2}, \mu_{C.R2}\} = \text{Min}\{\mu_{y.R2}, \mu_{d4}\} = 0,78$.

Правило R3: $\mu_{c3} = \text{Min}\{\mu_{y.R3}, \mu_{C.R3}\} = \text{Min}\{\mu_{y.R3}, \mu_{d3}\} = 0,22$.

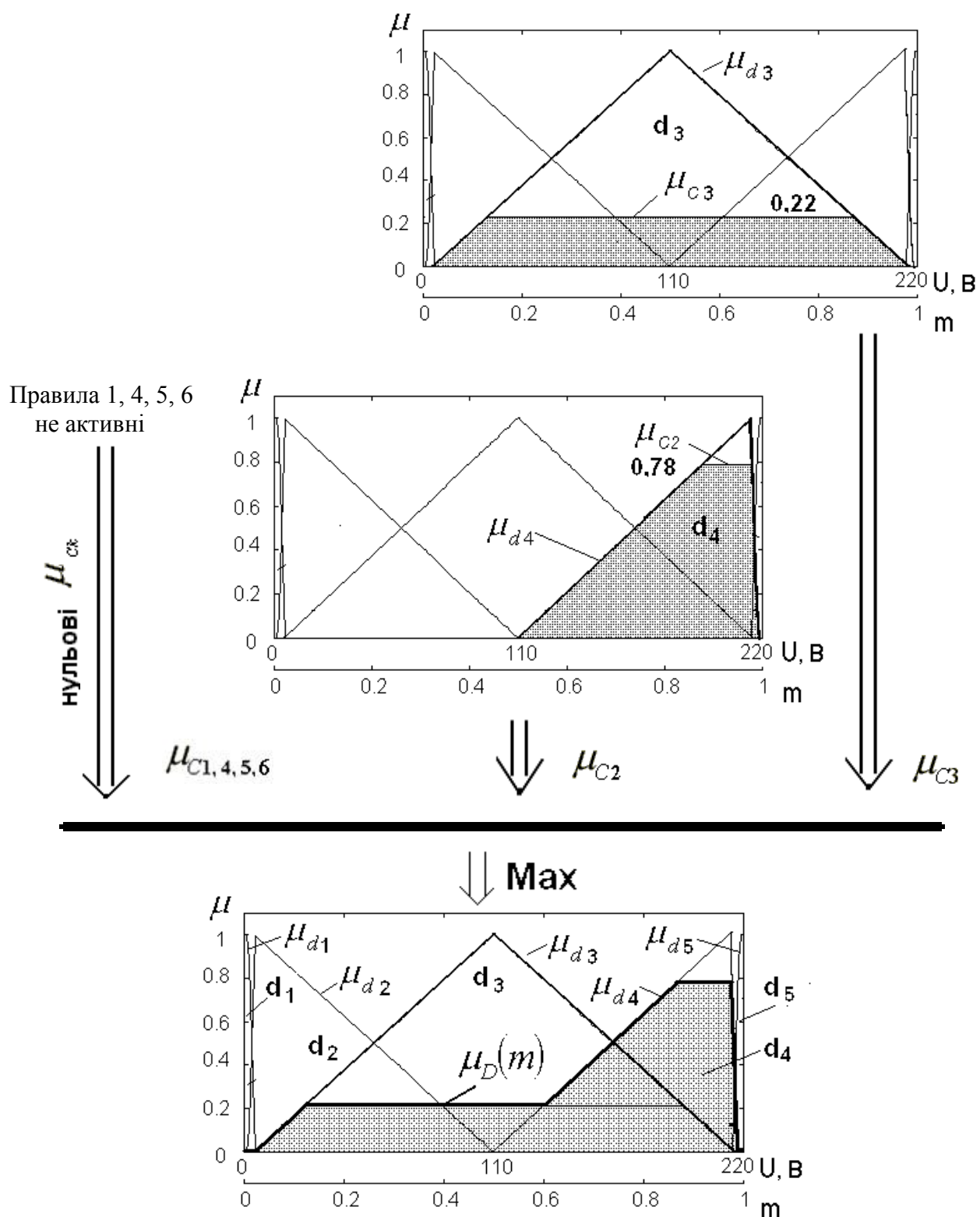
Правило R2:



Правило R3:



7. Графічна інтерпретація процесу накопичення висновків правил за логічного рішення (аккумуляція)



4.10 Прямий і зворотний методи нечіткого висновку

За аналогією зі звичайними продукційними системами важливим компонентом систем нечітких продукцій є так званий метод, або схема формування висновків на основі нечітких умов у базі правил. Найбільш відомими є два методи формування висновків: прямий і зворотний.

Прямий метод формування висновків у системах нечітких продукцій, який ще називають також метод нечіткого вихідного формування, або метод прямого нечіткого ланцюжка міркувань (fuzzy forward-chaining reasoning; метод, що підтверджує), заснований на використанні нечіткого узагальнення правила модус поненс – FMP (fuzzy modus ponens, нечіткий модус поненс – MP). Згідно з Л. Заде суть нечіткого модус поненса полягає в такому, що класична імплікація $A \supset B$ (з A випливає B) у правилі MP замінюється на правило нечіткої продукції: «ЯКЩО $x \in \mathcal{A}$, ТО $y \in \mathcal{B}$ », де $\mathcal{A}$ і $\mathcal{B}$ – нечіткі множини, а саме правило нечіткої продукції є деяким нечітким відношенням між змінними x і y , у цьому разі $x \in X$ і $y \in Y$. Що стосується посилення A правила MP, то воно замінюється на нечітку умову « $x \in \mathcal{A}'$ », де $\mathcal{A}'$ – нечітка множина, що відображає знання про реальне значення змінної x . Об'єднання правила нечіткої продукції і нечіткої умови дає змогу одержати нову інформацію про значення змінної y в формі « $y \in \mathcal{B}'$ ». У цьому разі висновок за правилом FMP формується як функція належності нечіткої множини $\mathcal{B}'$ на основі функції належності умови $\mathcal{A}'$ і функції належності нечіткої імплікації як відповідного нечіткого відношення з використанням одного з методів нечіткої композиції.

Стосовно системи нечітких продукцій прямий метод висновку реалізується за допомогою перетворення окремих фактів проблемної області в конкретні значення функцій належності умов нечітких продукцій. Після цього перетворення за одним з методів нечіткої композиції знаходяться значення функцій належності висновків правих частин за кожним із правил. Ці значення функцій належності або є шуканим результатом висновку, або можуть бути використані як додаткові умови у розглянутій базі правил. У цьому разі правила, які можуть бути використані для виконання нечіткої композиції, називають *активними*.

Процес формування висновку прямим методом у системах нечітких продукцій у загальному випадку може мати рекурсивний (ітеративний) характер. Він може бути зупинений або у випадку відсутності активних правил нечітких продукцій, або у випадку одержання функції належності висновку, який є цільовим у контексті рішення вихідної проблеми.

Зворотний метод висновку, або метод зворотного нечіткого ланцюжка міркувань (fuzzy backward-chaining reasoning; метод, що заперечує) оснований на використанні нечіткого узагальнення правила модус толленс – FMT (fuzzy

modus tollens, нечіткий модус толленс – МТ). Суть нечіткого модус толленса полягає в такому, що класична імплікація $A \supset B$ у правилі МТ замінюється на правило нечіткої продукції: «ЯКЩО $x \in \mathcal{A}$, ТО $y \in \mathcal{B}$ », де $\mathcal{A}$ і $\mathcal{B}$ – нечіткі множини, а правило нечіткої продукції є деяким нечітким відношенням між змінними x і y , у цьому разі $x \in X$ і $y \in Y$, як і в методі FMP. Висновок $\mathcal{B}$ замінюється нечітким висновком у формі «чи є $y \in \mathcal{B}'$?» або « $y \in \mathcal{B}'$?». У цьому разі нечітка множина $\mathcal{B}'$ не дорівнює нечіткій множині $\mathcal{B}$, використаній у висновку правила нечіткої продукції. Метою висновку методу зворотного нечіткого ланцюжка міркувань є встановлення істинності умови правил нечіткої продукції у формі «чи є $x \in \mathcal{A}'$?» або « $x \in \mathcal{A}'$?» (якщо встановлено, що $\mathcal{B}$ є хибним, то і $\mathcal{A}$ є також хибним). У цьому випадку висновок за правилом FMT виходить як функція належності нечіткої множини $\mathcal{A}'$ на основі функції належності висновку $\mathcal{B}'$ і функції належності нечіткої імплікації як відповідного нечіткого відношення з використанням одного з методів нечіткої композиції.

Принципова відмінність між зворотними методами виведення висновків у нечітких і звичайних системах продукцій полягає в тому, що стосовно нечітких систем функції належності умов невідомі і мають бути якимось чином задані. Процес зворотного виведення в системах нечітких продукцій починається з підстановки окремих значень функції належності висновків у праві частини відповідних правил нечітких продукцій, які в цьому випадку стають активними. Після аналізу кожного з активних правил знаходяться функції належності умов, які в подальшому використовуються в цих правилах.

Отримані функції належності умов беруться як підумови, які можуть бути використані як функції належності нових висновків у розглянутій базі правил нечітких продукцій.

Процес висновку зворотним методом також має рекурсивний (ітеративний) характер. Він може бути зупинений або у випадку відсутності нових активних правил, або у випадку одержання значень функцій належності умов, які підтверджуються фактами проблемної області. Подібне підтвердження умов характеризує успіх процесу виведення і справедливість значень функції належності вихідних висновків.

4.11 Основні етапи нечіткого логічного висновку

Нечітка логіка широко використовується для управління технічними пристроями і процесами. Розробка та застосування систем нечіткого висновку містить у собі ряд етапів, реалізація яких виконується за допомогою розглянутих раніше основних положень нечіткої логіки.

Інформацією, яка надходить на вхід системи нечіткого висновку, є отримані з датчиків параметри. Ці параметри відповідають реальним змінам

процесу управління. Інформація, яка формується на виході системи нечіткого висновку, відповідає вихідним змінним процесу управління.

Системи нечіткого висновку призначені для перетворення значень вхідних змінних процесу управління у вихідні змінні на основі використання нечітких правил. Для цього системи нечіткого висновку мають містити базу правил нечітких продукцій і реалізовувати нечіткий висновок на основі посилянь або умов, представлених у формі нечітких лінгвістичних висловлювань.

Для отримання висновків у системах нечіткого логічного рішення запропоновані декілька алгоритмів, які базуються на розподілі процесу висновку на ряд послідовних етапів. Таким чином можливо не тільки досягнути певної систематизації розглянутих понять нечіткої логіки, а й отримати деяку загальну схему, на базі якої можна створювати індивідуальні алгоритми нечіткого висновку.

Основними етапами *нечіткого висновку* є такі дії (рис. 4.27):

1. Формування бази правил систем нечіткого висновку.
2. Фаззифікація вхідних змінних.
3. Агрегування підумов у нечітких правилах.
4. Активізація або композиція висновків у нечітких правилах.
5. Акумуляування висновків нечітких правил продукцій.

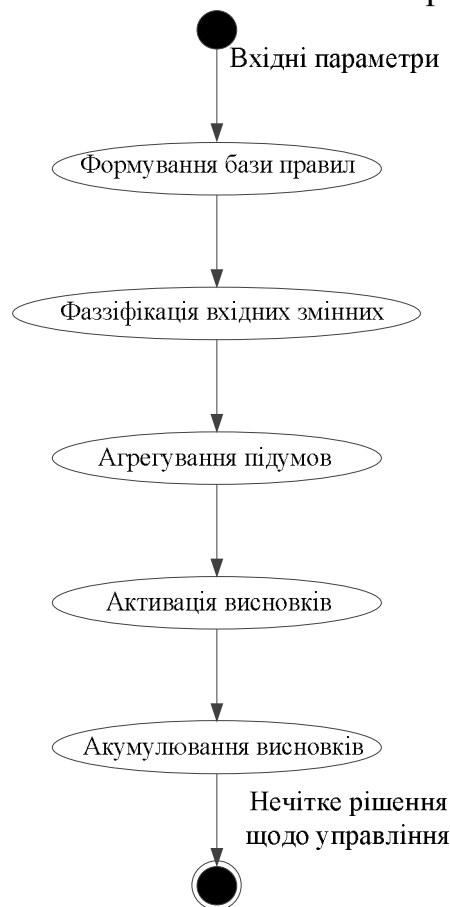


Рис. 4.27. Діаграма дій за формуванням висновку

4.11.1 Формування бази правил систем нечіткого висновку

База правил систем нечіткого висновку призначена для формального наведення емпіричних знань або знань експертів у тій чи іншій проблемній сфері. У системах нечіткого висновку використовуються правила нечітких продукцій, у яких умови і висновки сформульовані в термінах нечітких лінгвістичних висловлювань. Сукупність таких правил будемо далі називати базами нечітких знань – нечітких продукцій.

Розглянемо правила нечітких продукцій, що в максимальному наближенні відповідають алгоритму висновку Мамдані, який найбільш широко практично застосовується для систем нечіткого моделювання.

Розглянемо формування бази нечітких правил.

База нечітких правил є кінцевою множиною правил нечітких продукцій, погоджених щодо використаних у них лінгвістичних змінних. Найбільш часто база правил наводиться у формі структурованого тексту:

ПРАВИЛО _1: ЯКЩО «Умова_1» ТО «Висновок_1» (F_1)

ПРАВИЛО _2: ЯКЩО «Умова_2» ТО «Висновок_2» (F_2)

...

...

ПРАВИЛО _n: ЯКЩОУ «Умова_n» ТО «Висновок_n» (F_n)

або в еквівалентній формі:

RULE_1: IF Condition_1 THEN Conclusion_1(F_1)

RULE_2: IF Condition_2 THEN Conclusion_2 (F_2)

...

...

RULE_n: IF Condition_n THEN Conclusions (F_n)

Через F_i ($i \in \{1, 2, \dots, n\}$) позначені *коефіцієнти визначеності* або *вагові коефіцієнти* відповідних правил. Ці коефіцієнти можуть набувати значень з інтервалу $[0, 1]$. У випадку, якщо ці вагові коефіцієнти відсутні, зручно вважати, що їх значення на рівні 1.

Погодженість правил щодо використаних лінгвістичних змінних означає, що умови і висновки правил сформовані нечіткими лінгвістичними висловлюваннями, і для кожного з нечітких висловлювань визначені функції належності терм-множини для кожної з лінгвістичних змінних.

4.11.2 Фаззіфікація (Fuzzification)

У контексті нечіткої логіки під фаззіфікацією розуміють не тільки окремий етап виконання нечіткого висновку, але й власне процес або процедуру знаходження значень функцій належності нечітких множин (термів) на основі звичайних (не нечітких) вхідних даних. Фаззіфікацію ще називають *вступом до нечіткості*.

Метою етапу фаззіфікації є встановлення відповідності між конкретним (звичайно – числовим) значенням окремої базової вхідної змінної (параметра) системи нечіткого висновку і значенням функції належності відповідного їй терма вхідної лінгвістичної змінної. Після завершення цього етапу для всіх вхідних змінних мають бути визначені конкретні значення функцій належності за кожним з лінгвістичних термів, які використовуються в підумовах бази правил системи нечіткого висновку.

Формально процедура фаззіфікації виконується в такий спосіб. До початку цього етапу передбачаються відомими конкретні значення всіх вхідних змінних системи нечіткого висновку, тобто множина значень $U_i = \{u_1, u_2, \dots, u_m\}$. У загальному випадку кожне $u_i \in U_i$, де U_i – універсум лінгвістичної змінної β_i . Ці значення можуть бути отримані або від датчиків, або деяким іншим зовнішнім стосовно системи нечіткого висновку способом.

Далі розглядається кожна з підумов виду « $\beta_i \in \alpha_i$ » правил системи нечіткого висновку, де α_i – деякий терм із відомою функцією належності $\mu(u)$. У цьому разі значення u_i , використовується як аргумент $\mu(u)$, тим самим знаходиться кількісне значення $b_i' = \mu(u)$. Це значення і є результатом фаззіфікації підумови « $\beta_i \in \alpha_i$ ».

Етап фаззіфікації вважається закінченим, коли будуть знайдені всі значення $b_i' = \mu(c_i)$ для кожної з підумов усіх правил, що входять до розглянутої бази правил системи нечіткого висновку. Цю множину значень позначимо через $V = \{b_i'\}$. У випадку, коли деякий терм α_i лінгвістичної змінної β_i , не присутній у жодному з нечітких висловлювань (є нульовим), то відповідне йому значення функції належності не використовується в процесі фаззіфікації.

Якщо в деякій підумові зустрічається терм із модифікатором, то процедура фаззіфікації виконується аналогічним чином відповідно до функції належності терма після виконання операції, відповідної до такого модифікатора.

Приклад 4.4

Для ілюстрації виконання цього етапу розглянемо процес фаззіфікації трьох нечітких висловлювань: «напруга є низька», «напруга є середня», «напруга є висока» для вхідної лінгвістичної змінної β_1 (напруга джерела живлення) універсуму U . Їм відповідають нечіткі висловлювання першого виду: « $\beta_1 \in \alpha_1$ », « $\beta_1 \in \alpha_2$ », « $\beta_1 \in \alpha_3$ ».

Припустимо, що поточна напруга дорівнює 55 В, тобто $u=55$ В.

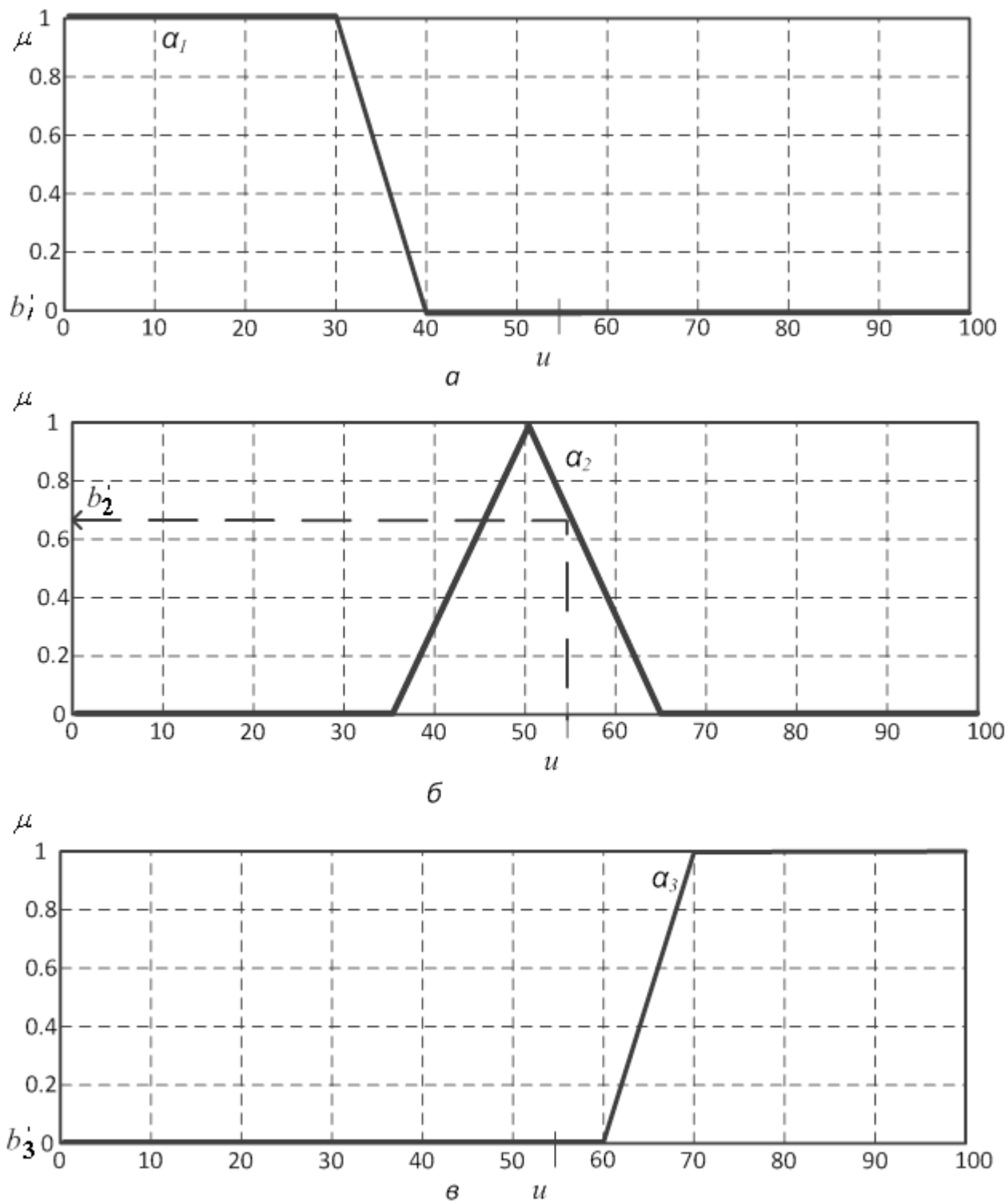


Рис. 4.28. Приклад фаззифікації вхідної лінгвістичної змінної «напруга» для трьох нечітких висловлювань: *a* – «напруга є низька»; *б* – «напруга є середня»; *в* – «напруга є висока»

Фаззифікація першого нечіткого висловлювання дає в результаті число 0, яке означає його ступінь істинності і отримане підстановкою значення $u=55$ В у якості аргументу функції належності терма α_1 (рис. 4.28, *a*). Фаззифікація

другого нечіткого висловлювання дає в результаті число 0,67 (наближене значення), яке визначає його ступінь істинності та отримане підстановкою значення $u=55$ В у якості аргументу функції належності терма a_2 (рис. 4.28, б). Фаззифікація третього нечіткого висловлювання дає в результаті число 0, яке визначає його ступінь істинності та отримане підстановкою значення $u=55$ В у якості аргументу функції належності терма a_3 (рис. 4.28, в).

4.11.3 Агрегування (Aggregation)

Агрегування є процедурою визначення ступеня істинності умов за кожним із правил системи нечіткого висновку.

Формально процедура агрегування виконується в такий спосіб. До початку цього етапу передбачаються відомими значення істинності всіх підумов системи нечіткого висновку, тобто множина значень $B = \{b_i'\}$. Далі розглядається кожна з умов правил системи нечіткого висновку. Якщо умова правила є нечітким висловлюванням виду 1 або 2 (розділ 4.8.1), то ступінь його істинності дорівнює відповідному значенню b_i' .

Якщо ж умова складається з декількох підумов виду, причому лінгвістичні змінні в підумовах попарно не ідентичні одна одній, то визначається ступінь істинності складного висловлювання на основі відомих значень істинності підумов. У цьому разі для визначення результату дії сполучника «І» може бути використана операція нечіткої кон'юнкції, а для визначення результату дії сполучника «АБО» може бути використана операція нечіткої диз'юнкції. У цьому разі значення b_i' використовуються як аргументи відповідних логічних операцій. Саме так знаходяться кількісні значення істинності всіх умов правил системи нечіткого висновку.

Приклад 4.5

Для ілюстрації виконання цього етапу розглянемо приклад процесу агрегування двох нечітких висловлювань: «струм є середній» І «напруга є висока» і «струм є середній» АБО «напруга є висока» для вхідної лінгвістичної змінної β_1 – струм $j_i \in J_i$, де J_i – універсум лінгвістичної змінної β_1 та β_2 – напруга $u_i \in U_i$, де U_i – універсум лінгвістичної змінної β_2 .

Припустимо, що поточний струм дорівнює 55 А, тобто $j_1=55$ А, а напруга дорівнює $u_2=70$ В. Тоді агрегування першого нечіткого висловлювання з використанням операції нечіткої кон'юнкції (*min*) (рис. 4.29) дає в результаті число $b_1''=0,67$ (наближене значення), яке визначає його ступінь істинності і виходить як мінімальне зі значень 0,67 і 0,8 (рис. 4.29, а). Агрегування другого нечіткого висловлювання з використанням операції нечіткої диз'юнкції (*max*)

дає в результаті число $b_1''=0,8$, яке визначає його ступінь істинності й виходить як максимальне зі значень 0,67 і 0,8 (рис. 4.29, б).

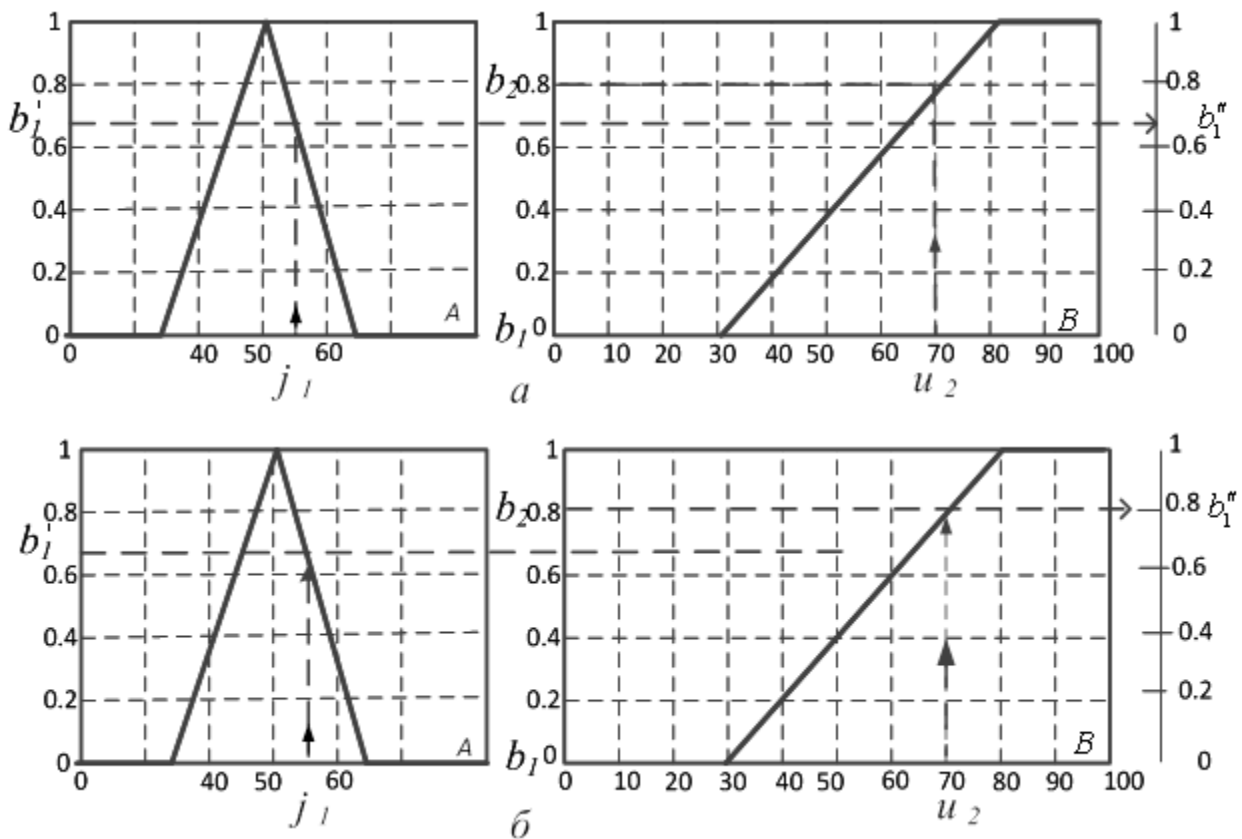


Рис. 4.29. Приклади агрегування підумов для двох нечітких висловлювань:
а – «струм є середній» І «напруга є висока»; б – «струм є середній» АБО «напруга є висока»

4.11.4 Активація (Activation) – імплікація

Активація в системах нечіткого висновку є процедурою або процесом знаходження ступеня істинності кожного з висновків нечітких правил.

У випадку формування бази правил системи нечіткого висновку задаються вагові коефіцієнти F_i для кожного правила (за замовчуванням передбачається, якщо ваговий коефіцієнт не заданий явно, його значення на рівні 1).

Формально процедура активізації виконується в такий спосіб. До початку цього етапу передбачаються відомими значення істинності всіх умов системи нечіткого висновку, тобто множина значень $B'' = \{b_1'', b_2'', \dots, b_n''\}$ і значення вагових коефіцієнтів F_i для кожного правила. Далі розглядається кожен з висновків правил бази знань. Якщо висновок правила є нечітким

висловлюванням виду 1 або 2, то ступінь його істинності дорівнює алгебраїчному добутку відповідного значення b_i на ваговий коефіцієнт F_i .

Якщо ж висновок складається з декількох підвисновків виду 3, причому лінгвістичні змінні в підвисновках попарно не рівні один одному, то ступінь істинності кожного з підвисновків дорівнює алгебраїчному добутку відповідного значення b_i на ваговий коефіцієнт F_i . Таким чином, знаходяться усі значення c_k ступенів істинності підвисновків для кожного із правил, що входять в базу правил системи нечіткого висновку. 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. З цією метою можна використовувати один з методів, що є модифікацією того або іншого методу нечіткої композиції:

- **Min**–активізація:

$$\mu'(y) = \min\{c_i, \mu(y)\};$$

- **Prod**–активізація:

$$\mu'(y) = c_i \cdot \mu(y);$$

- **Average**–активізація:

$$\mu'(y) = 0.5(c_i + \mu(y)),$$

де $\mu(y)$ – функція належності терма, який є значенням деякої вихідної змінної ω_j , заданої на універсумі Y .

Етап активізації вважається закінченим, коли для кожної з вихідних лінгвістичних змінних, що входять в окремі підвисновки нечітких правил, будуть визначені функції належності нечітких множин їх значень, тобто сукупність нечітких множин: $c_1, c_2, \dots, c_q$, де q – загальна кількість підвисновків у базі правил системи нечіткого висновку.

Приклад 4.6

Для ілюстрації виконання цього етапу розглянемо приклад процесу активізації висновку в наступному нечіткому правилі фазового управління вмиканням тиристора:

ЯКЩО «струм є середній», ТО «кут є великий».

Вхідною лінгвістичною змінною в цьому правилі є β_1 – струм, а вихідною змінною є β_2 – кут вмикання тиристора.

Припустимо, що поточний струм у колі має значення 55 А, тобто $j_I=55$ (рис. 4.30).

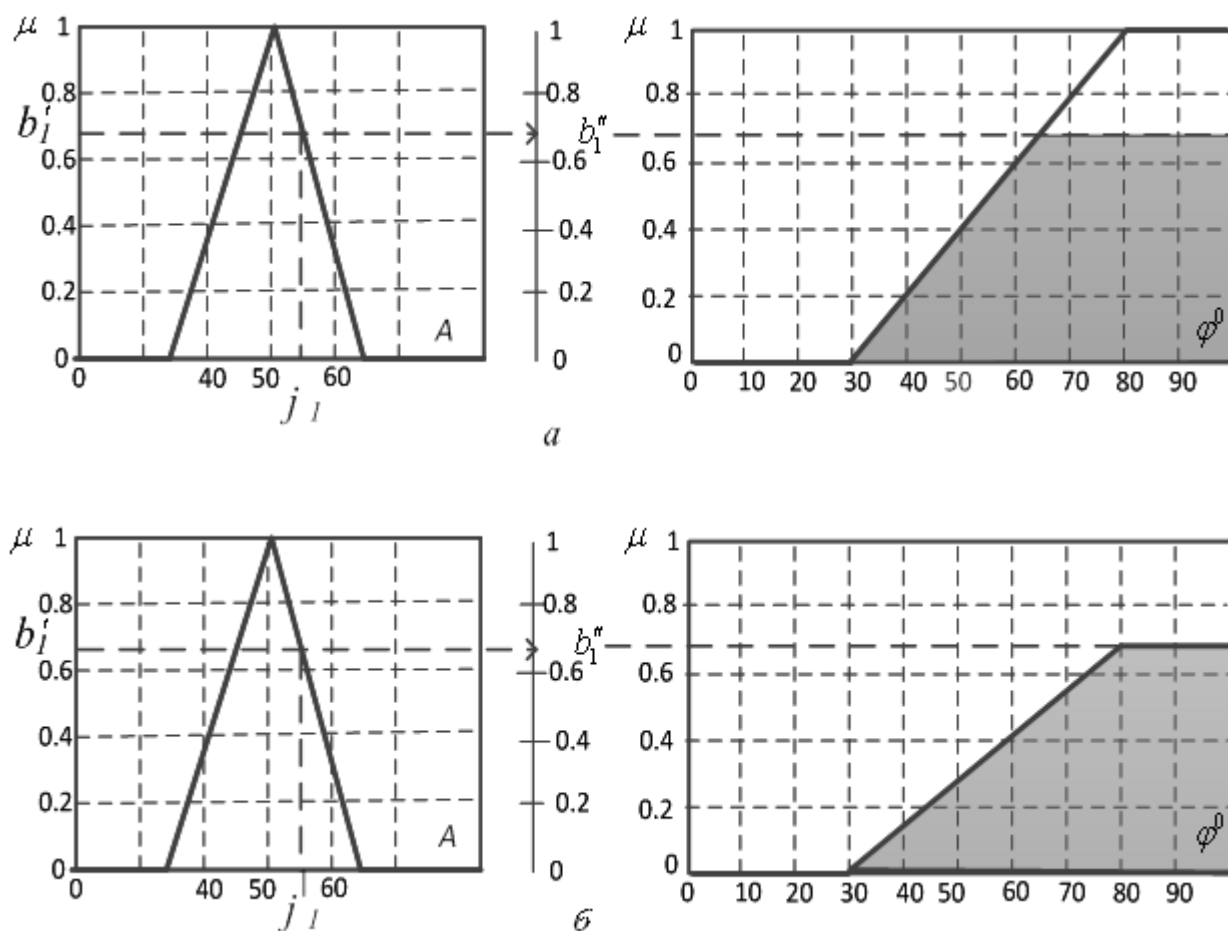


Рис. 4.30. Приклад активізації висновку: *a* – методом Min-активізації; *б* – методом Prod-активізації

Оскільки агрегування умови цього правила дає в результаті $b_I''=0,67$, а ваговий коефіцієнт дорівнює 1 (за замовчуванням), то значення 0,67 буде використовуватися як c_I для одержання результату активізації. Результат, отриманий методом min-активізації, зображений на рис. 4.30, *a* сірим кольором; результат, отриманий методом prod-активізації, зображений на рис. 4.30, *б* сірим кольором.

4.11.5 Акумуляція (Accumulation)

Акумуляція або *акumuлювання* в системах нечіткого висновку є процедурою або процесом знаходження функції належності для кожної з вихідних лінгвістичних змінних множин $W=\{\omega_1, \omega_2, \dots, \omega_s\}$.

Мета акумуляції полягає в тому, щоб об'єднати або акумулювати всі ступені істинності висновків (підвисновків) для одержання функції належності кожної з вихідних змінних. Причина необхідності виконання цього етапу полягає в тому, що підвисновки, які відносяться до однієї і тієї ж вихідної лінгвістичної змінної, належать різним правилам системи нечіткого висновку.

Формально процедура акумуляції виконується в такий спосіб. До початку цього етапу передбачаються відомими значення істинності всіх висновків для кожного із правил R_k , що входять у розглянуту базу правил P системи нечіткого висновку, у формі сукупності нечітких множин:

$$C_1, C_2, \dots, C_q,$$

де q – загальна кількість підвисновків у базі правил. Далі послідовно розглядається кожна з вихідних лінгвістичних змінних $\omega_j \in W$ та нечіткі множини, що відносяться до неї:

$$C_{j1}, C_{j2}, \dots, C_{jq}.$$

Результат акумуляції для вихідної лінгвістичної змінної ω_j визначається як об'єднання нечітких множин $C_{j1}, C_{j2}, \dots, C_{jq}$ за однією з формул об'єднання, що висвітлені раніше.

Етап акумуляції вважається закінченим, коли для кожної з вихідних лінгвістичних змінних будуть визначені підсумкові функції належності нечітких множин їх значень, тобто сукупність нечітких множин:

$$C'_1, C'_2, \dots, C'_s,$$

де s – загальна кількість вихідних лінгвістичних змінних у базі правил системи нечіткого висновку.

Приклад 4.7

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 «кут» вмикання тиристора у деякій системі нечіткого висновку.

Припустимо, що функції належності цих нечітких множин зображені на рис. 4.31, а, б, в відповідно.

Акумуляція цих функцій належності методом *max*–об'єднання нечітких множин C_{11}, C_{12}, C_{13} дає змогу одержати в результаті функцію належності вихідної лінгвістичної змінної «кут» вмикання тиристора, яка представлена на рис. 4.31, г. Контуром затемненої фігури і є функція належності, яка описує нечітку множину C'_1 вихідної лінгвістичної змінної ω_1 .

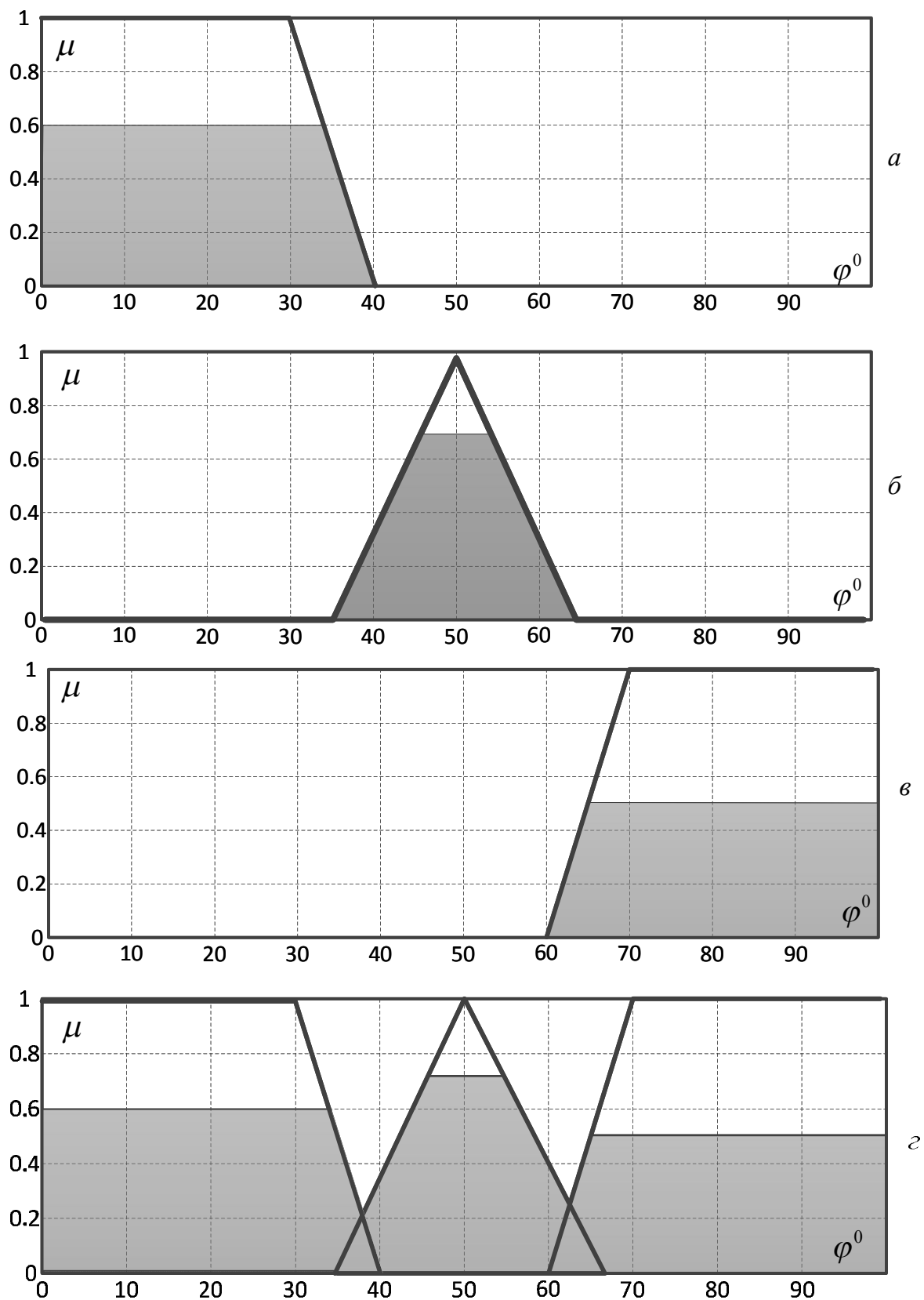


Рис. 4.31. Приклад акумуляції висновку для вихідної лінгвістичної змінної «кут» вмикання тиристора: *a* – нечітка множина C_{11} ; *б* – нечітка множина C_{12} ; *в* – нечітка множина C_{13} ; *г* – *max*-об'єднання нечітких множин C_{11} , C_{12} , C_{13}

Приклад 4.8

Розглянемо формування моделі акумуляції висновку у разі прийняття рішення щодо визначення чайових *tip* за двома вхідними параметрами: *сервіс* обслуговування *service* та *якість їжі* *food* – за обід у ресторані.

Вхідні параметри *сервіс* та *якість їжі* визначимо лінгвістичними змінними *service* та *food* і представимо гіпотетично нечіткими множинами, опишемо їх функціями належності в рейтинговому інтервалі від 0 до 10. Вихідну лінгвістичну змінну *tip* представимо гіпотетично нечіткими множинами, опишемо їх функціями належності в рейтинговому інтервалі від 0 до 30 %.

1. Сформуємо нечітку терм-множину вхідної лінгвістичної змінної гаусовими функціями належності *service* {*poor*, *good*, *excellent*}.

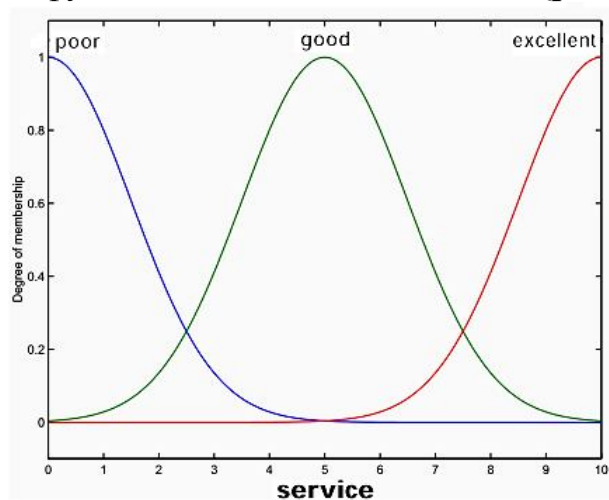


Рис. 4.32. Терм-множина змінної *service*

2. Сформуємо нечітку терм-множину вхідної лінгвістичної змінної *Z* та *S* функціями належності *food* {*rancid*, *delicious*}.

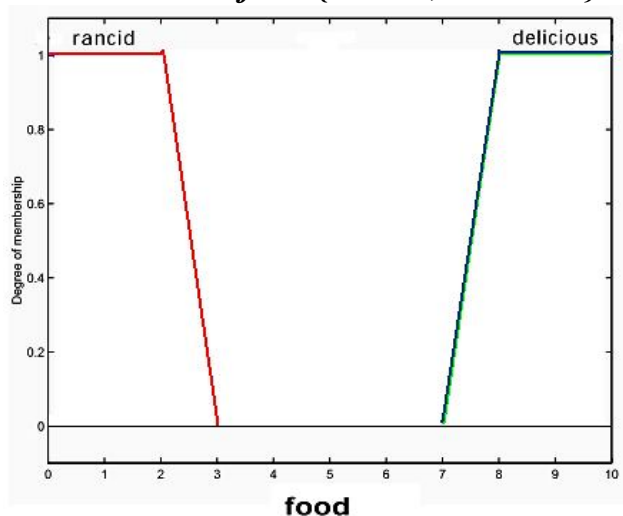


Рис. 4.33. Терм-множина змінної *food*

3. Сформуємо нечітку терм-множину вихідної лінгвістичної змінної *tip*{*cheap*, *average*, *generous*} трикутними функціями належності.

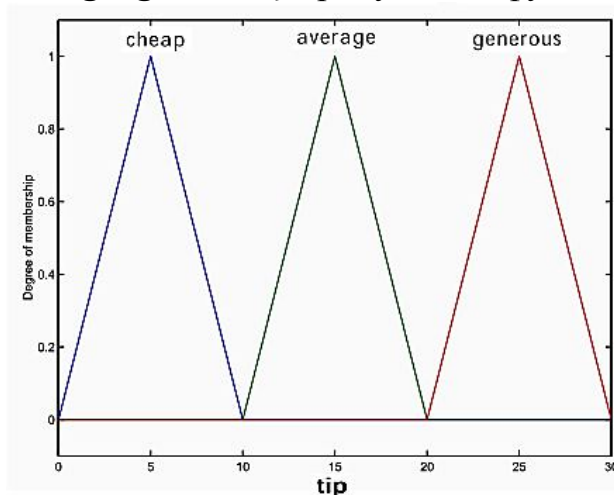


Рис. 4.34. Терм-множина змінної *tip*

4. Визначимо правила прийняття рішення щодо чайових у випадку використання однієї лінгвістичної змінної *service*.

1. *If service is poor, then tip is cheap*
2. *If service is good, then tip is average*
3. *If service is excellent, then tip is generous*

Доповнення бази правил за другою лінгвістичною змінною *food*.

4. *If food is rancid, then tip is cheap*
5. *If food is delicious, then tip is generous*

Проведемо оптимізацію правил бази знань.

1. *If service is poor or the food is rancid, then tip is cheap*
2. *If service is good, then tip is average*
3. *If service is excellent or food is delicious, then tip is generous*

5. Представимо структуру нечіткого логічного рішення графічно.

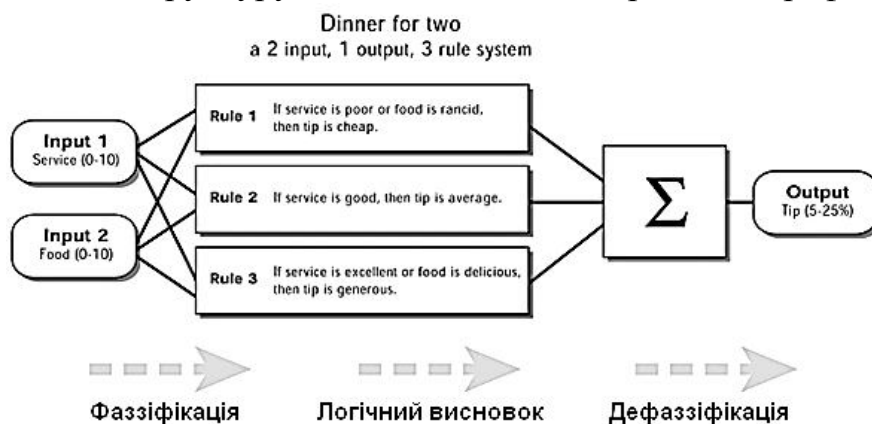


Рис. 4.35. Структура нечіткого логічного рішення

6. Виконаємо нечіткий логічний висновок за алгоритмом Мамдані.

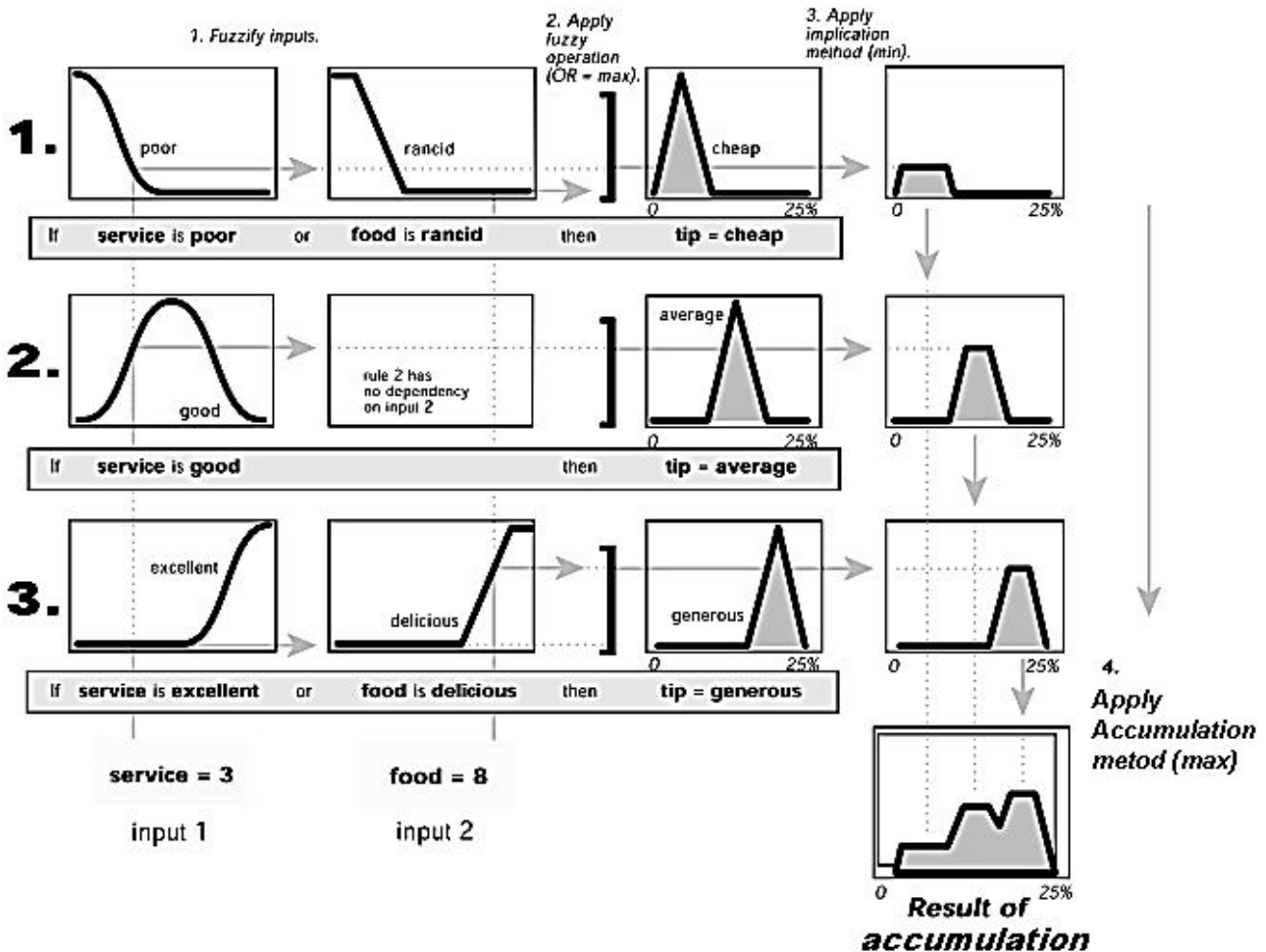


Рис. 4.36. Алгоритм логічного висновку

4.12 Дефаззифікація нечіткого логічного висновку

Дефаззифікація (defuzzification) в системах нечіткого висновку є процедурою або процесом знаходження реального фізичного (не нечіткого) значення для кожної з вихідних лінгвістичних змінних множини $W = \{\omega_1, \omega_2, \dots, \omega_s\}$.

Мета дефаззифікації полягає в тому, щоб, використовуючи результати акумуляції всіх вихідних лінгвістичних змінних, одержати звичайне кількісне значення (crisp value) кожної з вихідних змінних, тобто сигнал управління або рішення щодо управління, яке може бути використане спеціальними пристроями, зовнішніми по відношенню до системи нечіткого висновку.

У сучасних системах управління пристрої та механізми здатні сприймати традиційні команди у формі кількісних значень відповідних сигналів управління. Саме з цієї причини необхідно перетворити нечіткі множини в деякі конкретні значення змінних. Тому дефаззифікацію називають також приведенням до чіткості.

Формально процедура дефаззифікації виконується в такий спосіб. До початку цього етапу передбачаються відомими функції належності всіх вихідних лінгвістичних змінних у формі нечітких множин: $C_1', C_2', \dots, C_s'$, де s – загальна кількість вихідних лінгвістичних змінних у базі правил системи нечіткого висновку. Далі послідовно розглядаються кожна з вихідних лінгвістичних змінних $\omega_j \in W$ і нечітка множина, що відноситься до неї C_j' . Результат дефаззифікації для вихідної лінгвістичної змінної ω_j визначається у вигляді кількісного значення $y_j \in R$, одержаного за однією з математичних формул.

Етап дефаззифікації вважається закінченим, коли для кожної з вихідних лінгвістичних змінних будуть визначені підсумкові кількісні значення у формі деякого дійсного числа, тобто у вигляді $y_1, y_2, \dots, y_s$, де s – загальна кількість вихідних лінгвістичних змінних у базі правил системи нечіткого висновку.

Для виконання числових розрахунків на етапі дефаззифікації можуть бути використані такі основні математичні методи дефаззифікації: *центру ваги*, *центру ваги для одноточкових множин*, *центру площі*, *лівого модального значення*, *правого модального значення*.

4.12.1 Метод центру ваги

Центр ваги (Cog, COG, Centre of Gravity) або центроїд площі розраховується за формулою:

$$y = \frac{\int_{Min}^{Max} x \cdot \mu(x) dx}{\int_{Min}^{Max} \mu(x) dx}.$$

У формулі використовуються такі позначення: y – результат дефаззифікації; x – змінна, що відповідає вихідній лінгвістичній змінній ω ; $\mu(x)$ – функція належності нечіткої множини, що відповідає вихідній змінній ω після етапу акумуляції; Min і Max – ліва й права точки інтервалу носія нечіткої множини вихідної змінної ω .

У випадку дефаззифікації методом центру ваги звичайне (не нечітке) значення вихідної змінної дорівнює абсцисі центру ваги плоскої фігури, обмеженої графіком кривої функції належності відповідної вихідної змінної.

Графічна інтерпретація процедури дефаззифікації методом центру ваги функції належності вихідної лінгвістичної змінної «*кут*» *вмикання тиристора* зображено на рис. 4.37. У цьому випадку $y_1 = 37^\circ$ (приблизне значення). Це значення є точкою рівноваги однорідної плоскої фігури, обмеженої ламаною кривою функції належності вихідної лінгвістичної змінної та віссю абсцис.

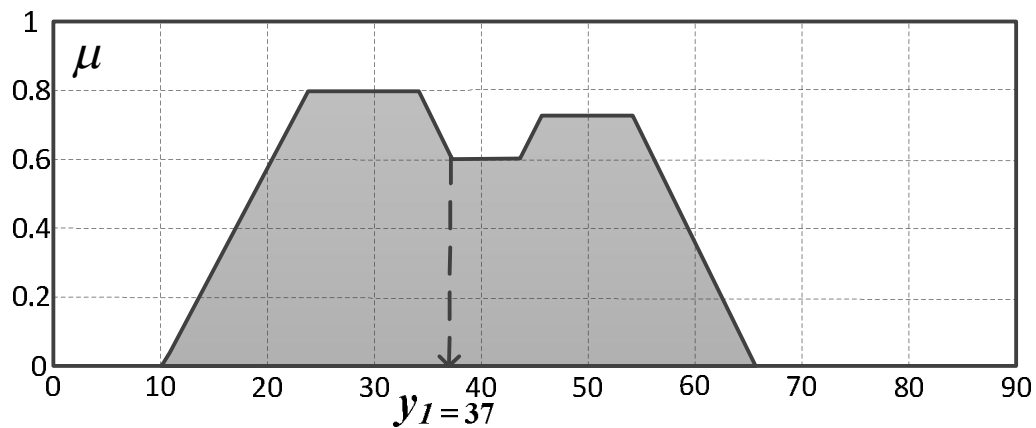


Рис. 4.37. Дефаззифікація вихідної лінгвістичної змінної «кут»
вмикання тиристора методом центру ваги

4.12.2 Метод центру ваги для одноточкових множин

Центр ваги (COGS, Centre of Gravity for Singletons) 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, кожне з яких характеризує єдине значення розглянутої вихідної лінгвістичної змінної.

Графічна інтерпретація процедури дефаззифікації методом центру ваги для одноточкових множин функції належності вихідної лінгвістичної змінної «кут» вмикання тиристора зображено на рис. 4.38. У цьому випадку приблизно $y_1 = 41^\circ$.

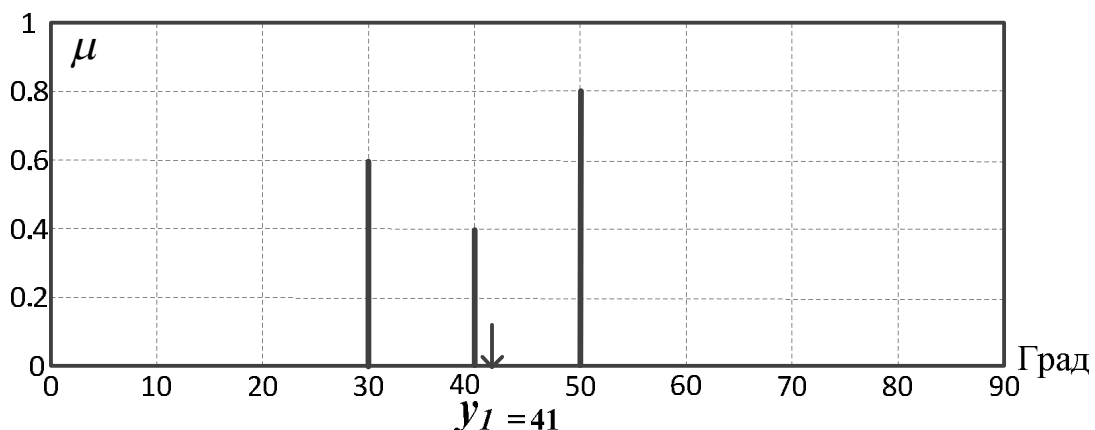


Рис. 4.38. Дефаззифікація лінгвістичної змінної «кут» вмикання тиристора методом центру
ваги для одноточкових множин

4.12.3 Метод центру площі

Центр площі (CoA, COA, Centre of Area, Bisector of Area) розраховується виходячи з рівняння $y=u$, де значення u визначається з виразу

$$\int_{Min}^u \mu(x)dx = \int_u^{Max} \mu(x)dx,$$

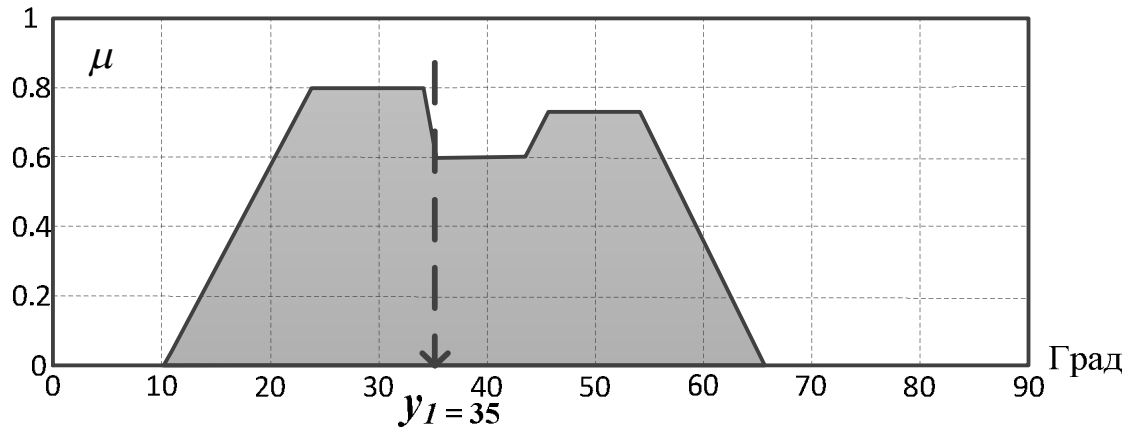


Рис. 4.39. Дефаззифікація вихідної лінгвістичної змінної «кут»
вмикання тиристора методом центру площі

Центр площі визначає точка на осі абсцис від перетину прямої паралельної осі ординат, яка ділить площу, обмежену графіком кривої функції належності відповідної вихідної змінної, на дві рівні частини, та осі абсцис. Іноді центр площі називають *бісектрисою площі*. Цей метод не може бути використаний у випадку одноточкових множин.

Графічна інтерпретація процедури дефаззифікації методом центру площі функції належності вихідної лінгвістичної змінної «кут» вмикання тиристора зображена на рис. 4.39. У цьому випадку $y_I=35^\circ$.

4.12.4 Метод лівого модального значення

Ліве модальне значення (LM, Left Most Maximum) розраховується за формулою:

$$y = \min\{x_m\},$$

де x_m – модальне значення (мода) нечіткої множини, що відповідає вихідній змінній ω після акумуляції.

Тобто значення вихідної змінної визначається як мода нечіткої множини для відповідної вихідної змінної або найменша з мод (сама ліва), якщо нечітка множина має кілька модальних значень.

Графічна інтерпретація процедури дефаззифікації методом лівого модального значення функції належності вихідної лінгвістичної змінної «кут» вмикання тиристора зображена на рис. 4.40. У цьому випадку $y_I=24^\circ$.

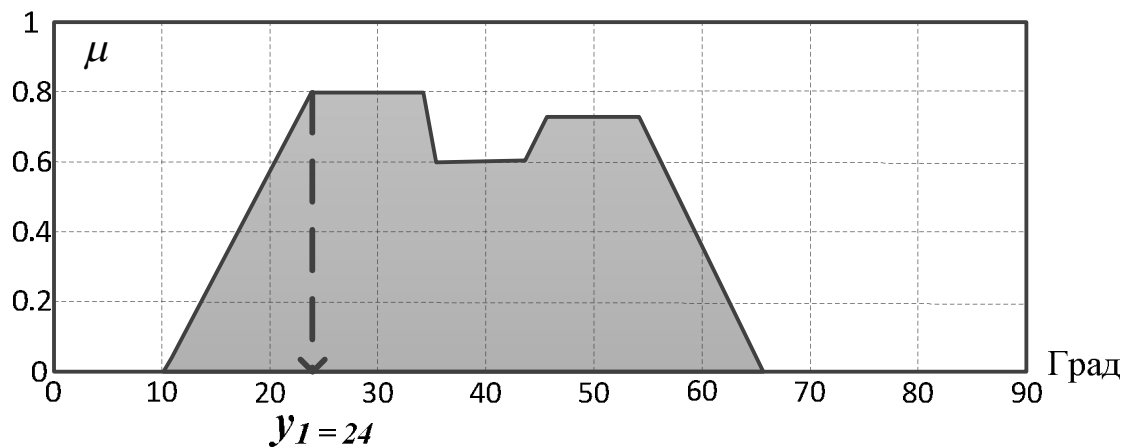


Рис. 4.40. Дефаззифікація вихідної лінгвістичної змінної «кут» вмикання тиристора методом лівого модального значення

4.12.5 Метод правого модального значення

Праве модальне значення (RM, Right Most Maximum) розраховується за формулою:

$$y = \max \{x_m\},$$

де x_m – модальне значення (мода) нечіткої множини для вихідної змінної ω після акумуляції.

У цьому випадку значення вихідної змінної також визначається як мода нечіткої множини для відповідної вихідної змінної або найбільша з мод (сама крайня права), якщо нечітка множина має кілька модальних значень.

Графічна інтерпретація процедури дефаззифікації функції належності вихідної лінгвістичної змінної «кут» вмикання тиристора зображена на рис. 4.41. У цьому випадку $y_I = 54^\circ$.

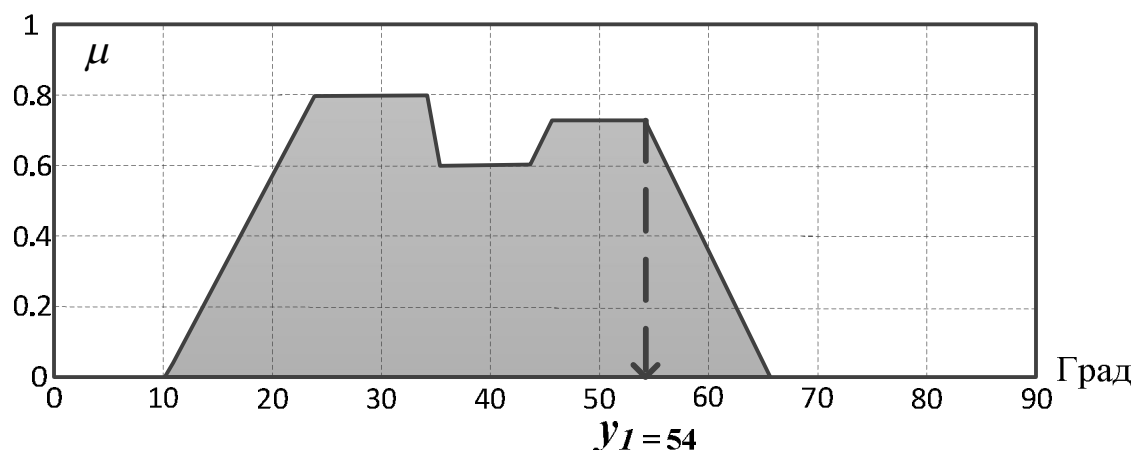


Рис. 4.41. Дефаззифікація вихідної лінгвістичної змінної «кут» вмикання тиристора методом правого модального значення

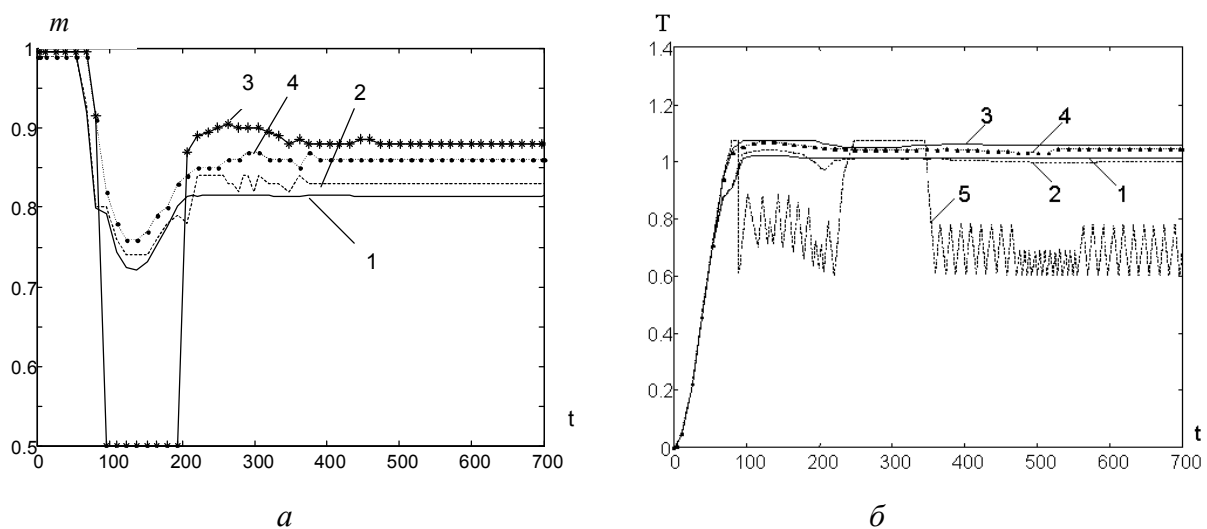
Приклад 4.9

Дефаззифікація нечіткого логічного рішення щодо формування коефіцієнта регулювання напруги m в електропечі.

На рис. 4.42 представлено передатні характеристики НЛК та регулятора напруги, отримані за нечіткого логічного рішення щодо формування коефіцієнта напруги m для різних методів дефаззифікації при наборі електропечію максимальної температури T (у відносних одиницях):

$$m = \frac{n \cdot T_M}{T_P} = \frac{n}{50},$$

де T_P – період регулювання 1 секунда; T_M – період змінної напруги; n – кількість періодів напруги в інтервалі дискретизації.



1 – метод центру ваги; 2 – метод центру площі; 3 – метод середнього модального значення; 4 – метод лівого модального значення; 5 – метод правого модального значення.

Рис. 4.42. Передатні характеристики для різних методів дефаззифікації у разі набору електропечію максимальної температури T : a – НЛК;
 $б$ – регулятора напруги

4.13 Основні алгоритми нечіткого висновку

Етапи нечіткого висновку можуть бути реалізовані неоднозначним чином, оскільки містять у собі окремі параметри, які мають бути фіксовані або специфіковані. Тим самим вибір конкретних варіантів параметрів кожного з етапів визначає деякий алгоритм, який у повному обсязі реалізує нечіткий висновок в інтелектуальних системах прийняття рішень. Розглянемо алгоритми, найбільш поширені в системах нечіткого висновку, зокрема, в енергетиці.

4.13.1 Алгоритм Мамдані (Mamdani)

Алгоритм Мамдані є одним з перших, який знайшов застосування в системах нечіткого висновку. Він був запропонований в 1975 р. англійським математиком Е. Мамдані (Ebrahim Mamdani) як метод для керування паровим двигуном. За своїм змістом цей алгоритм використовує розглянуті вище етапи, оскільки найбільш відповідає їхнім параметрам.

Формально *алгоритм Мамдані* може бути визначений у такій послідовності:

- **Формування бази правил систем нечіткого висновку.**
- **Фаззифікація вхідних змінних.**
- **Агрегування підумов у нечітких правилах.** Для знаходження ступеня істинності умов кожного із правил нечітких продукцій використовуються нечіткі логічні операції. Ті правила, ступіні істинності умов яких відмінні від нуля, вважаються *активними* і використовуються для подальших розрахунків.
- **Активізація підвисновків у нечітких правилах.** Для скорочення часу висновку враховуються тільки активні правила нечітких продукцій.
- **Акумуляція висновків нечітких правил.** Для об'єднання нечітких множин, відповідних до термів підвисновків.
- **Дефаззифікація вихідних змінних.** Традиційно використовується метод центру ваги або метод центру площі.

4.13.2 Алгоритм Цукамото (Tsukamoto)

Формально *алгоритм Цукамото* може бути визначений у такий спосіб:

- **Формування бази правил систем нечіткого висновку.**
- **Фаззифікація вхідних змінних.** Особливості фаззифікації збігаються з розглянутим алгоритмом Mamdani.
- **Агрегування підумов у нечітких правилах.** Для знаходження ступеня істинності умов усіх правил нечітких продукцій використовуються нечіткі логічні операції. Ті правила, ступінь істинності умов яких відмінний від нуля, вважаються *активними* й використовуються для

подальших розрахунків.

- **Активізація підвисновків у нечітких правилах.** Здійснюється аналогічно алгоритму Мамдані, після чого визначаються звичайні (не нечіткі) значення всіх вихідних лінгвістичних змінних у кожному з підвисновків активних правил нечітких. У цьому випадку значення вихідної лінгвістичної змінної w_j у кожному з підвисновку знаходиться як рішення рівняння

$$c_i = \mu(w_j) \quad (\forall i \in \{1, 2, \dots, q\}),$$

де q – загальна кількість підвисновків у базі правил.

- **Акумуляція висновків нечітких правил.** Фактично відсутня, оскільки розрахунки здійснюються зі звичайними дійсними числами w_j .
- **Дефаззифікація вихідних змінних.** Використовується модифікований варіант у формі методу центру ваги для одноточкових множин:

$$y = \frac{\sum_{i=1}^n c_i \cdot w_i}{\sum_{i=1}^n c_i},$$

де n – загальна кількість активних правил, у підвисновках яких присутня вихідна лінгвістична змінна w_j .

4.13.3 Алгоритм Ларсена (Larsen)

Формально *алгоритм Ларсена* може бути визначений у такий спосіб:

- **Формування бази правил систем нечіткого висновку.** Особливості формування бази правил аналогічні попереднім методам.
- **Фаззифікація вхідних змінних.** Особливості фаззифікації також аналогічні попереднім методам.
- **Агрегування підумов у нечітких правилах.** Використовуються попарно нечіткі логічні операції для знаходження ступеня істинності умов усіх правил нечітких продукцій (як правило, max–диз'юнкція і min–кон'юнкція). Правила, ступінь істинності умов яких відмінна від нуля, вважаються *активними* і використовуються для подальших розрахунків.
- **Активізація підвисновків у нечітких правилах.** Здійснюється використанням формули prod–активації та знаходження сукупності нечітких множин: $C_1, C_2, \dots, C_q$, де q – загальна кількість підвисновків у базі правил.
- **Акумуляція висновків нечітких правил.** Здійснюється за формулою

об'єднання нечітких множин, відповідних до термів підвисновків, що відносяться до одних і тих же самих вихідних лінгвістичних змінних.

- **Дефаззифікація вихідних змінних.** Може використовуватися будь-якою з розглянутих вище методів дефаззифікації.

4.13.4 Алгоритм Сугено (Sugeno)

Формально *алгоритм Сугено*, запропонований вченими Сугено і Такаги, може бути визначений у такий спосіб:

- **Формування бази правил систем нечіткого висновку.** У базі правил використовуються тільки правила нечітких продукцій у формі:
ПРАВИЛО <#>: ЯКЩО « $\beta_1 \in a_1'$ » І « $\beta_2 \in a_2'$ » ТО « $w = \varepsilon_1 \cdot \alpha_1 + \varepsilon_2 \alpha_2$ »,
де $\varepsilon_1, \varepsilon_2$ – деякі вагові коефіцієнти. У такому випадку значення вихідної змінної w у висновку визначається як деяке дійсне число.
- **Фаззифікація вхідних змінних.** Особливості фаззифікації збігаються з розглянутими вище алгоритмами.
- **Агрегування підумов у нечітких правилах продукцій.** Для знаходження ступеня істинності умов усіх правил нечітких продукцій, як правило, використовується логічна операція Min–кон'юнкція. Ті правила, ступінь істинності умов яких відмінна від нуля, вважаються *активними* і використовуються для подальших розрахунків.
- **Активація підвисновків у нечітких правилах продукцій.** По–перше, з використанням методу min–активації знаходяться значення ступенів істинності всіх висновків нечітких правил. По–друге, здійснюється розрахунок звичайних (чітких) значень вихідних змінних кожного правила. Це виконується з використанням формули, що наведена у висновку правила, в яку замість a_1 і a_2 підставляються значення вхідних змінних до виконання фаззифікації. Тим самим визначаються множина значень $C = \{c_1, c_2, \dots, c_n\}$ і множина значень вихідних змінних $W = \{w_1, w_2, \dots, w_n\}$, де n – загальна кількість правил у базі знань.
- **Акумуляція висновків нечітких правил.** Фактично відсутня, оскільки розрахунки здійснюються зі звичайними дійсними числами w_j .
- **Дефаззифікація вихідних змінних.** Використовується модифікований варіант у формі методу центру ваги для одноточкових множин.

4.13.5 Спрощений алгоритм нечіткого висновку

Формально *спрощений алгоритм* може бути визначений у такий спосіб:

- **Формування бази правил систем нечіткого висновку.** У базі правил використовуються тільки правила нечітких продукцій у формі:
ПРАВИЛО <#>: ЯКЩО « $\beta_1 \in a_1'$ » И « $\beta_2 \in a_2'$ » ТО « $w = \varepsilon$ »,

де ε – деяке дійсне число;

- **Фаззифікація вхідних змінних.** Особливості фаззифікації збігаються з розглянутими вище у разі опису даного етапу;
- **Агрегування підумов у нечітких правилах.** Для знаходження ступеня істинності умов усіх нечітких правил, як правило, використовується логічна операція $\min$ –кон’юнкція. Ті правила, ступінь істинності умов яких відмінна від нуля, вважаються *активними* і використовуються для подальших розрахунків;
- **Активація підвисновків у нечітких правилах.** Здійснюється з використанням методу $\min$ –активації, за допомогою чого знаходять значення ступенів істинності всіх висновків нечітких правил $C = \{c_1, c_2, \dots, c_n\}$, де n – загальна кількість правил у базі знань;
- **Акумуляція висновків нечітких правил.** Фактично відсутня, оскільки розрахунки здійснюються зі звичайними дійсними числами c_j ;
- **Дефаззифікація вихідних змінних.** Використовується модифікований варіант у формі методу центру ваги для одноточкових множин.

У разі вирішення практичних завдань нечіткого моделювання можуть одночасно використовуватися кілька алгоритмів нечіткого висновку з метою одержання найбільш адекватних результатів.

Приклад 4.10

Розглянемо графічну інтерпретацію виконання нечіткого логічного висновку за **алгоритмом Мамдані** (рис. 4.43) щодо визначення чайових **tip** за двома параметрами – сервіс обслуговування **service** та якість їжі **food** – за обід у ресторані (приклад 4.8).

Приклад 4.11

Розглянемо графічну інтерпретацію виконання нечіткого логічного висновку за **алгоритмом Сугено** (рис. 4.44) щодо визначення чайових **tip** за двома параметрами – сервіс обслуговування **service** та якість їжі **food** – за обід у ресторані (приклад 4.8) за наступних значень вихідної лінгвістичної змінної $\text{tip}\{5, 15, 25\}$.

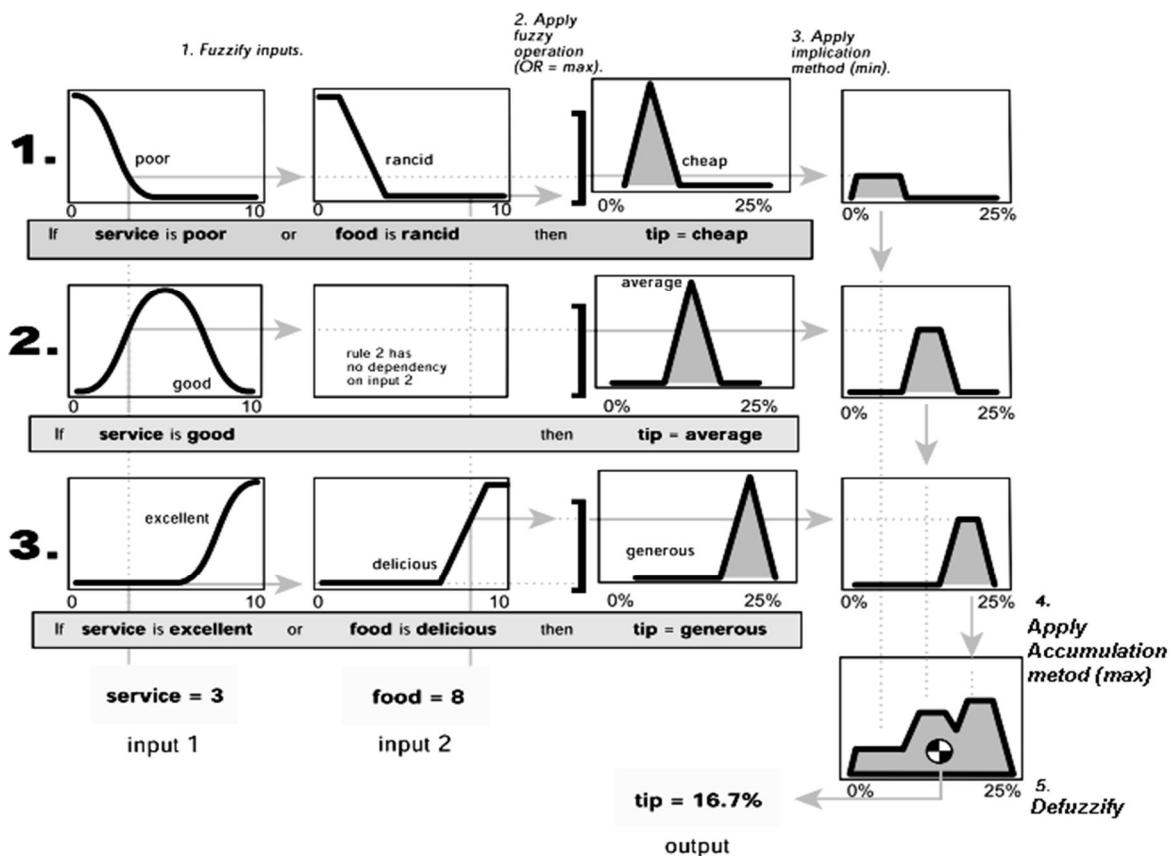


Рис. 4.43. Нечіткий логічний висновок за алгоритмом Мамдані

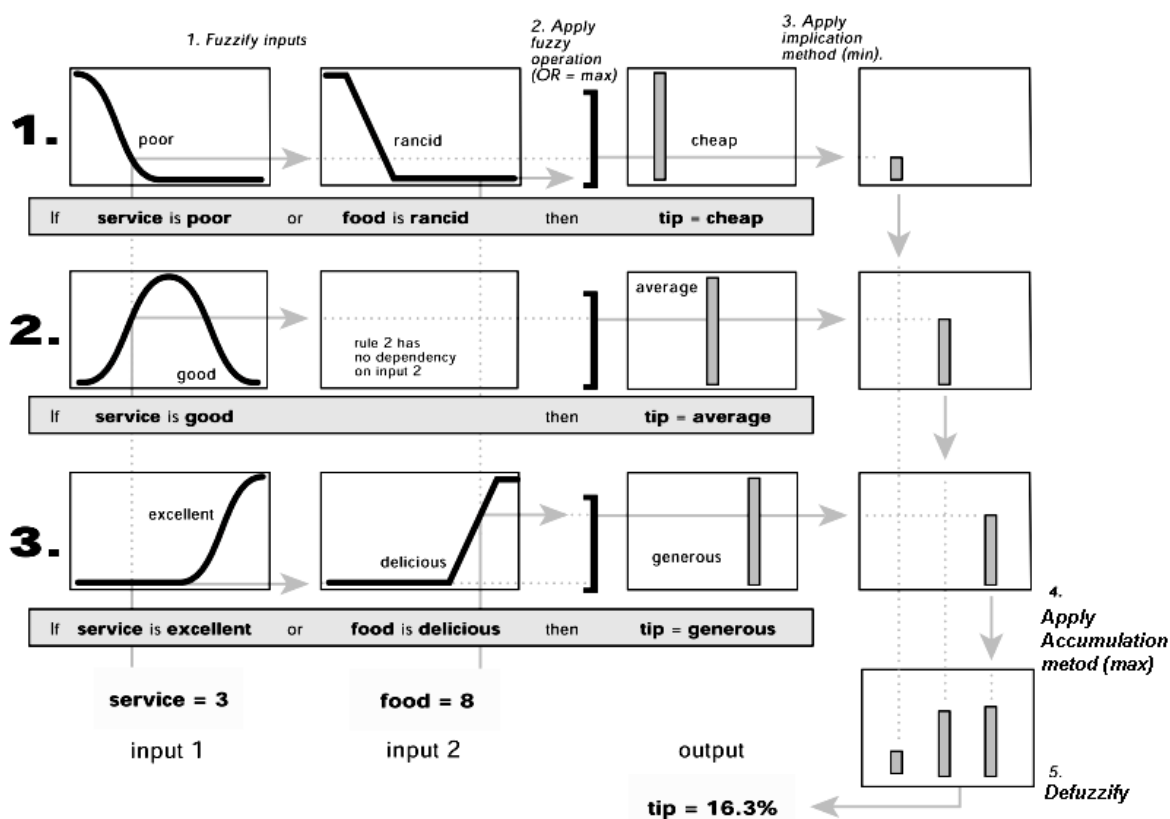


Рис. 4.44. Нечіткий логічний висновок за алгоритмом Сугено

На рис.4.45 представлена поверхня відгуку НЛК для нечіткого логічного рішення за алгоритмом Сугено.

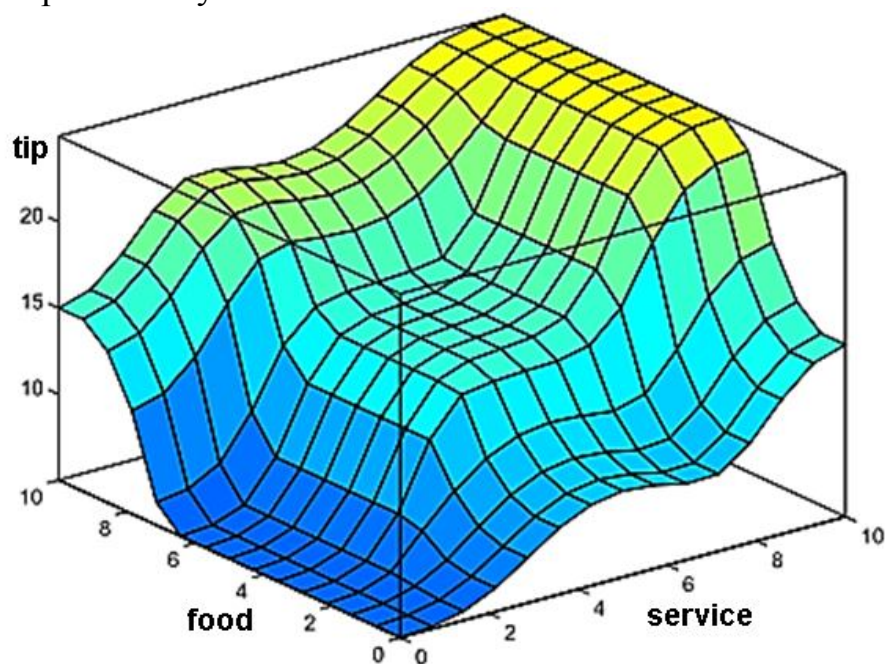


Рис. 4.45. Поверхня відгуку НЛК для нечіткого логічного рішення за алгоритмом Сугено

Приклад 4.12

Визначення пріоритету переведення розподільної електричної мережі напругою 6 кВ на напругу 20 кВ за алгоритмом Мамдані на основі дев'яти параметрів: сумарна довжина ліній 6 кВ; період функціонування; середня довжина фідерів мережі; навантаження; тенденція зміни навантаження; територіальна щільність навантаження; індекс середньої тривалості вимикань; індекс середньої частоти вимикань; населення району електропостачання.

За основу синтезу моделі критерію переведення розподільної мережі на клас напруги 20 кВ і процедури її реалізації приймемо числове значення пріоритету переведення розподільної мережі на напругу 20 кВ на основі нечіткого лінгвістичного аналізу множини параметрів мережі 6(10) кВ.

Синтез лінгвістичної моделі критерію доцільності переведення мережі на напругу 20 кВ полягає у визначенні впливу кожного з параметрів, аналізі комбінацій та кореляції їх спільного впливу і відтворенні його в правилах бази знань нечіткого логічного контролера, формуванні процедури прийняття рішення щодо пріоритету переведення мережі і коригування правил бази знань на основі імітаційного моделювання прийняття рішення.

Кожний з параметрів мережі змінюється в певному діапазоні. Тому необхідно визначити дійсні межі їхньої зміни, виходячи з нормативної документації, досвіду та імітаційного математичного моделювання.

Аналіз параметрів мережі є складним та інтелектуальним процесом осмислення інформації про прийняті для аналізу вхідні параметри, зіставлення її з даними, отриманими з різних джерел. Результат аналізу впливає на формування правил впливу значень параметрів і характеру їх зміни на прийняття рішення щодо рівня пріоритету. За відсутності достовірного математичного опису впливу параметра на значення пріоритету, ми не в змозі розрахувати його конкретне значення. У такому випадку встановлюємо лінгвістичне поле достовірності пріоритету, тобто межі нечіткої множини діапазону його зміни і можливого характеру цієї зміни.

Створення нечіткого логічного контролера проводиться шляхом лінгвістичного представлення діапазонів зміни вхідних параметрів нечіткими змінними та їх математичним описом функціями належності, які в правилах відтворюють вплив параметра на вихідні лінгвістичні нечіткі змінні пріоритету, які також математично описані функціями належності. Оскільки ми задаємося мінімальним і максимальним значеннями показника пріоритету, то результатом дефазифікації є числове значення пріоритету переведення мережі на напругу 20 кВ, що знаходиться в заданих межах.

Модель критерію P переведення мережі на 20 кВ на основі нечіткого логічного рішення представляється як деяка функція F від стану мережі

$$S = \{line_6kv, time_net, length_net, load, tendency, density, SAIDI, SAIFI, population\},$$

яка ставить у відповідність кожному стану деяке дійсне число пріоритету p з діапазону $[0, 10]$, через функцію належності $\mu(p)$ нечіткого логічного висновку:

$$P = F(S): P \rightarrow \{\mu(p)/p\}, p \in [0, 10]. \quad (4.1)$$

Для аналізу стану мережі S дев'ять параметрів представимо лінгвістичними змінними з відповідними термами:

- сумарна довжина ліній 6 кВ – *line_6kv*;
- період функціонування – *time_net*;
- середня довжина фідерів мережі – *length_net*;
- навантаження – *load*;
- тенденція зміни навантаження – *tendency*;
- територіальна щільність навантаження – *density_MW/km2*;
- індекс середньої тривалості вимикань – *SAIDI*;
- індекс середньої частоти вимикань – *SAIFI*;
- населення району електропостачання – *population*.

На фізичних діапазонах (універсамах) лінгвістичних змінних (у відносних одиницях) формуємо нечіткі множини з відповідними термами та описуємо їх функціями належності.

Лінгвістична змінна «сумарна довжина ліній 6 кВ» – *line_6kv*, представлена двома нечіткими змінними: «значна» з термом *s* та «незначна» – *ns*: $line_6kv = \{s, ns\}$.

Нечіткі змінні *s* (типу Z) та *ns* (типу S) представлені трапецієподібними сплайн-функціями належності (рис. 4.46):

$$\mu(x; a, b, c, d) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ 1, & b \leq x \leq c \\ \frac{d-x}{d-c}, & c \leq x \leq d \\ 0, & d \leq x \end{cases}, \quad (4.2)$$

де $a = -0,5$; $b = -0,3$; $c = 0$; $d = 1$ – для $\mu(ns)$,
 $a = 0$; $b = 1$; $c = 1,3$; $d = 1,5$ – для $\mu(s)$.

Лінгвістична змінна «період функціонування» – *time_net* представлена трьома нечіткими змінними (рис. 4.47):

«короткий» з термом *k*, «середній» – *m* та «довгий» – *l*: $time_net = \{k, m, l\}$.

Нечіткі змінні *k* (типу Z) та *l* (типу S) представлені трапецієподібними сплайн-функціями належності (4.2) з відповідними коефіцієнтами: $a = -0,5$; $b = -0,3$; $c = 0$; $d = 1$ – для $\mu(k)$ та $a = 0,4$; $b = 0,8$; $c = 1,3$; $d = 1,5$ – для $\mu(l)$.

Нечітка змінна *m* представлена трикутною функцією належності

$$\mu_{\Delta}(x; a, b, c) = \begin{cases} \frac{x-a}{b-a}, & a \leq x \leq b \\ \frac{c-x}{c-b}, & b \leq x \leq c \\ 0, & c \leq x \end{cases}, \quad (4.3)$$

де $a = 0$; $b = 0,4$; $c = 0,8$ – для $\mu(m)$.

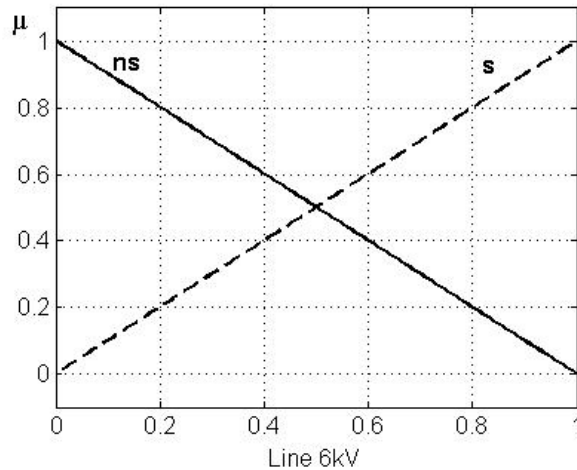


Рис. 4.46. Функції належності лінгвістичної змінної *line_6kv*

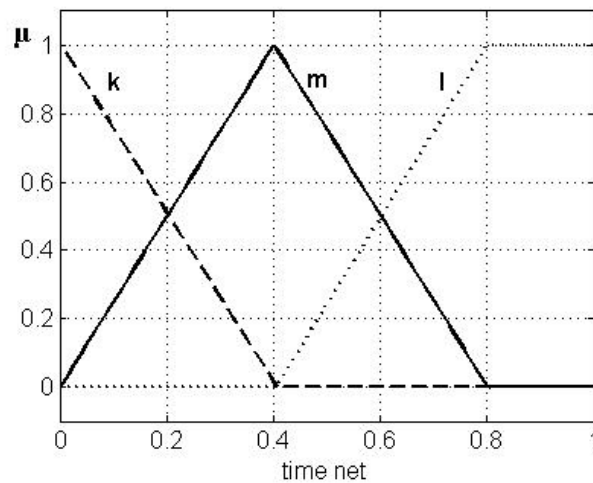


Рис. 4.47. Функції належності лінгвістичної змінної *time_net*

Лінгвістична змінна «середня довжина фідерів середньої напруги» – *length_net* представлена трьома нечіткими змінними (рис. 4.48): «мала» з термом *sm*, «середня» – *m* та «велика» – *l*: $length_net = \{sm, m, l\}$.

Нечіткі змінні *sm* (типу Z) та *l* (типу S) представлені трапецієподібними сплайн-функціями належності (4.2) з відповідними коефіцієнтами: $a = -0,5$; $b = -0,3$; $c = 0$; $d = 0,5$ – для $\mu(sm)$ та $a = 0,5$; $b = 1$; $c = 1,3$; $d = 1,5$ – для $\mu(l)$.

Нечітка змінна *m* представлена трикутною функцією належності (4.3) з коефіцієнтами: $a = 0$; $b = 0,5$; $c = 1$ – для $\mu(m)$.

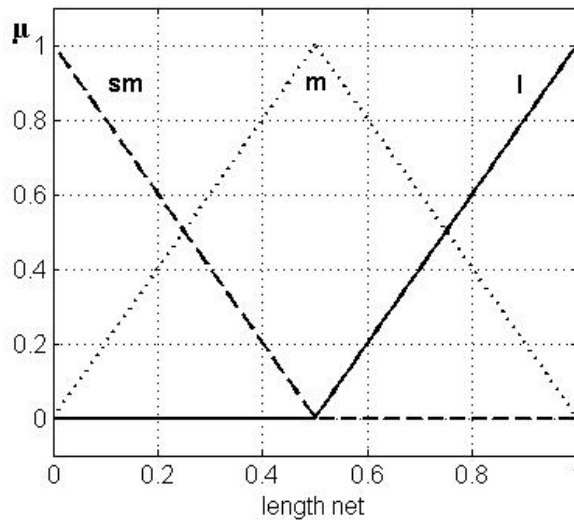


Рис. 4.48. Функції належності лінгвістичної змінної *length_net*

Лінгвістична змінна «навантаження» – *load* представлена трьома нечіткими змінними (рис. 4.49): «низька» з термом *low*, «нормальна» – *n* та «висока» – *h*: $load = \{low, n, h\}$.

Нечіткі змінні *low* (типу Z) та *h* (типу S) представлені трапецієподібними сплайн-функціями належності (4.2) з відповідними коефіцієнтами: $a = -0,5$; $b = -0,3$; $c = 0$; $d = 0,5$ – для $\mu(low)$ та $a = 0,5$; $b = 1$; $c = 1,3$; $d = 1,5$ – для $\mu(h)$.

Нечітка змінна *n* представлена трикутною функцією належності (4.3) з коефіцієнтами: $a = 0$; $b = 0,5$; $c = 1$ – для $\mu(n)$.

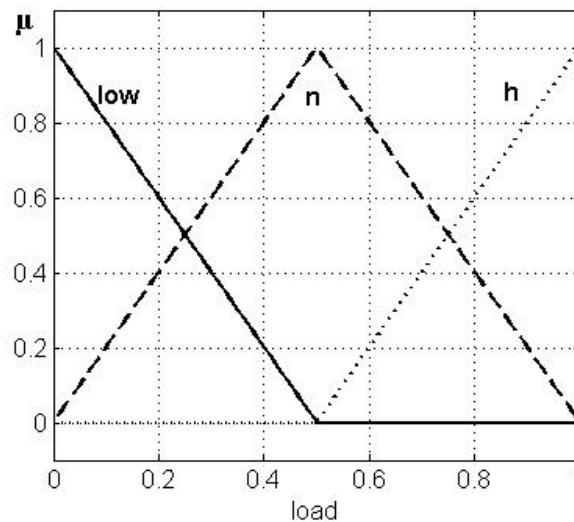


Рис. 4.49. Функції належності лінгвістичної змінної *load*

Лінгвістична змінна «тенденція зміни навантаження» – *tendency* представлена трьома нечіткими змінними (рис. 4.50): «негативна» з термом *n*, «нульова» – *zero* та «позитивна» – *p*: $tendency = \{n, zero, p\}$.

Нечіткі змінні *n* (типу *Z*) та *p* (типу *S*) представлені трапецієподібними сплайн-функціями належності (4.2) з відповідними коефіцієнтами: $a = -1,5$; $b = -1,3$; $c = -1$; $d = 0$ – для $\mu(n)$ та $a = 0$; $b = 1$; $c = 1,3$; $d = 1,5$ – для $\mu(p)$.

Нечітка змінна *zero* представлена трикутною функцією належності (4.3) з коефіцієнтами: $a = -0,1$; $b = 0$; $c = 0,1$ – для $\mu(zero)$.

Лінгвістична змінна «територіальна щільність навантаження» – $density_MW/km^2$ представлена двома нечіткими змінними (рис. 4.51): «низька» з термом *low* та «висока» – *high*: $density_MW/km^2 = \{low, high\}$.

Нечіткі змінні *low* (типу *Z*) та *high* (типу *S*) представлені трапецієподібними сплайн-функціями належності (4.2) з відповідними коефіцієнтами: $a = -0,5$; $b = -0,3$; $c = 0$; $d = 1$ – для $\mu(low)$ та $a = 0$; $b = 1$; $c = 1,3$; $d = 1,5$ – для $\mu(high)$.

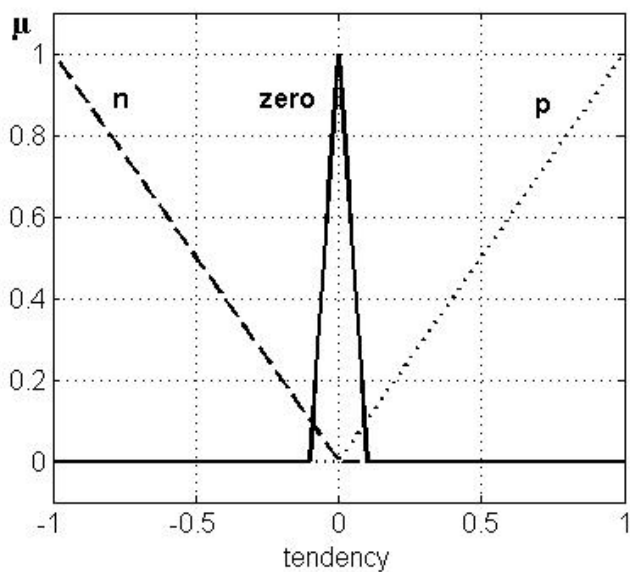


Рис. 4.50. Функції належності лінгвістичної змінної *tendency*

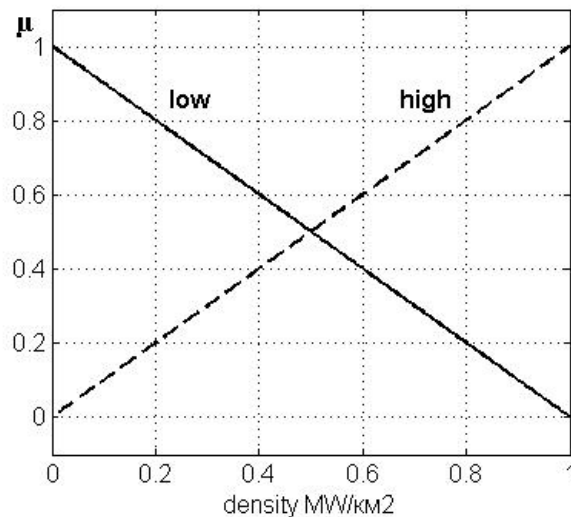


Рис. 4.51. Функції належності лінгвістичної змінної $density_MW/km2$

Лінгвістична змінна «індекс середньої тривалості вимикань» – $SAIDI$ представлена трьома нечіткими змінними (рис. 4.64): «нормальний» з термом n , «можливий» – pos та «критичний» – cr : $SAIDI = \{n, pos, cr\}$.

Нечіткі змінні n (типу Z) та cr (типу S) представлені трапецієподібними сплайн-функціями належності (4.2) з відповідними коефіцієнтами: $a = -0,5$; $b = -0,3$; $c = 0$; $d = 0,3$ – для $\mu(n)$ та $a = 0,3$; $b = 1$; $c = 1,3$; $d = 1,5$ – для $\mu(cr)$.

Нечітка змінна pos представлена трикутною функцією належності (4.3) з коефіцієнтами: $a = 0$; $b = 0,3$; $c = 1$ – для $\mu(pos)$.

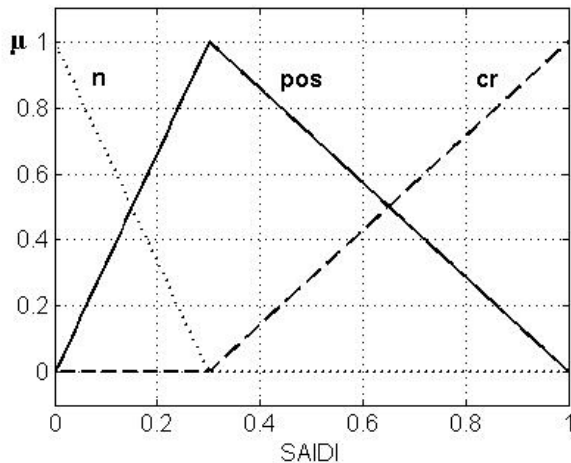


Рис. 4.52. Функції належності лінгвістичної змінної $SAIDI$

Лінгвістична змінна «індекс середньої частоти вимикань» – $SAIFI$ представлена двома нечіткими змінними (рис. 4.53): «низька» з термом low та «висока» – h : $SAIFI = \{low, h\}$.

Нечіткі змінні low (типу Z) та h (типу S) представлені трапецієподібними сплайн-функціями належності (4.2) з відповідними

коефіцієнтами: $a = -0,5$; $b = -0,3$; $c = 0$; $d = 1$ – для $\mu(low)$ та $a = 0$; $b = 1$; $c = 1,3$; $d = 1,5$ – для $\mu(h)$.

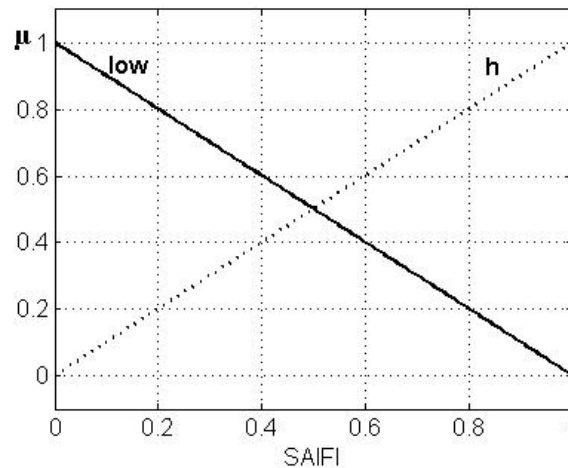


Рис. 4.53. Функції належності лінгвістичної змінної *SAIFI*

Лінгвістична змінна «населення району електропостачання» – *population*, представлена двома нечіткими змінними (рис. 4.54): «сільське» з термом *cs* та «міське» – *urban*: $population = \{cs, urban\}$.

Нечіткі змінні *cs* (типу *Z*) та *urban* (типу *S*) представлені трапецієподібними сплайн-функціями належності (4.2) з відповідними коефіцієнтами: $a = -0,5$; $b = -0,3$; $c = 0$; $d = 0,1$ – для $\mu(cs)$ та $a = 0$; $b = 1$; $c = 1,3$; $d = 1,5$ – для $\mu(urban)$.

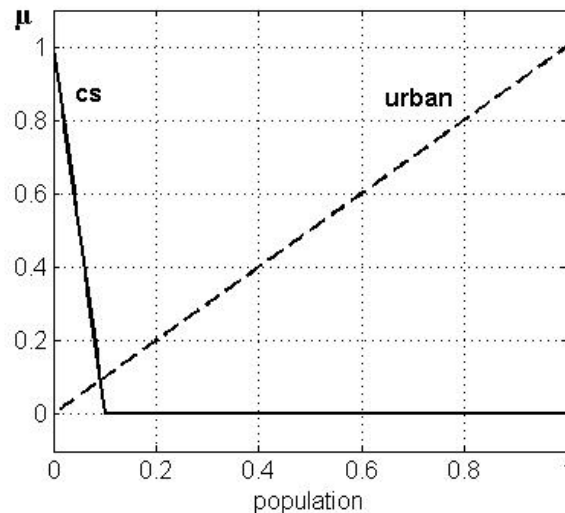


Рис. 4.54. Функції належності лінгвістичної змінної *population*

Вихідна лінгвістична змінна «пріоритет переведення мережі на напругу 20 кВ» – *priority* представлена множиною з одинадцяти нечітких числових змінних (рис. 4.55):

$$priority = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}.$$

Нечіткі змінні *0* (типу *Z*) та *10* (типу *S*) представлені трапецієподібними сплайн-функціями належності (4.2) з відповідними

коефіцієнтами: $a = -0,5$; $b = -0,3$; $c = 0$; $d = 1$ – для $\mu(0)$ та $a = 9$; $b = 10$; $c = 13$; $d = 15$ – для $\mu(10)$.

Нечіткі змінні з 1 по 9 представлені трикутними функціями належності (4.3) з коефіцієнтами:

- $a = 0$; $b = 1$; $c = 2$ – для $\mu(1)$;
- $a = 1$; $b = 2$; $c = 3$ – для $\mu(2)$;
- $a = 2$; $b = 3$; $c = 4$ – для $\mu(3)$;
- $a = 3$; $b = 4$; $c = 5$ – для $\mu(4)$;
- $a = 4$; $b = 5$; $c = 6$ – для $\mu(5)$;
- $a = 5$; $b = 6$; $c = 7$ – для $\mu(6)$;
- $a = 6$; $b = 7$; $c = 8$ – для $\mu(7)$;
- $a = 7$; $b = 8$; $c = 9$ – для $\mu(8)$;
- $a = 8$; $b = 9$; $c = 10$ – для $\mu(9)$.

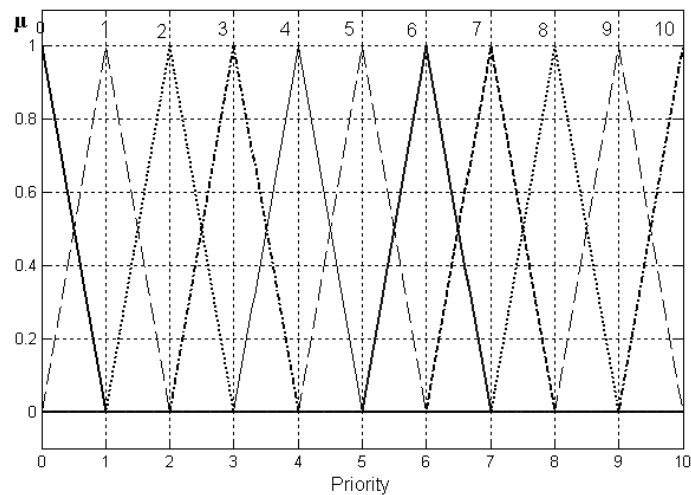


Рис. 4.55. Функції належності вихідної лінгвістичної змінної *priority*

Після визначення та описування лінгвістичних змінних формуються правила функціонування нечіткого логічного контролера. Для зручності представлення правил всі функції належності *mf* лінгвістичних змінних нумеруються зліва на право в межах діапазону кожної змінної, починаючи з одиниці. Вихідна база знань налічувала 3888 правил. У табл. 4.1 наведена початкова робоча база знань НЛК, отримана після першого етапу імітаційного моделювання контролера, яка складається з тридцяти правил формату:

ЯКЩО «умова з вхідних змінних», ТО «рішення – вихідна змінна» (вага).
Вхідні змінні в умовах правил об'єднані через логічний оператор «І» (AND).

Структурну схему нечіткого логічного контролера з описаними лінгвістичними змінними показано на рис. 4.56.

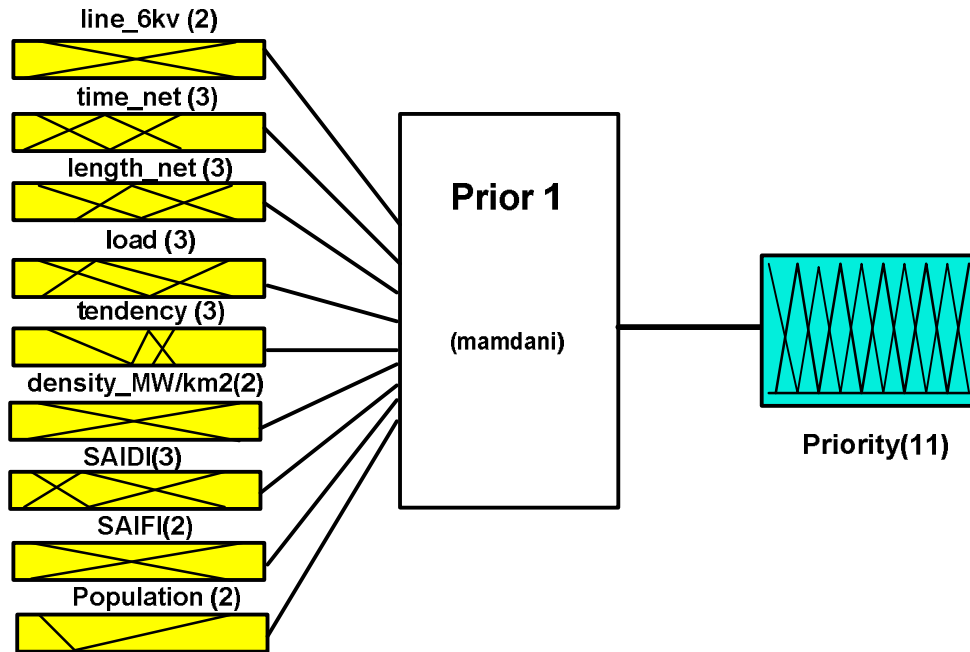


Рис. 4.56. Структурна схема нечіткого логічного контролера

Нечіткий логічний висновок може бути реалізований неоднозначно, оскільки містить у собі окремі процедури, які можуть реалізовуватися різними операторами. Тим самим вибір конкретних варіантів параметрів кожного з етапів визначає деякий алгоритм, який у повному обсязі реалізує нечіткий висновок у системах нечітких правил.

Формування логічного рішення проводилося за трьома етапами. Залежно від комбінації операторів на окремих етапах логічного рішення можна отримати різні стратегії логічного рішення.

Для виконання процедури нечіткого логічного рішення використано комбінацію операторів за алгоритмом Мамдані (Mamdani), яка наведена в табл. 4.2. У разі формування рішення щодо значення пріоритету, функції належності (ФН) нечітких змінних висновків правил обмежуємо ступенями виконання умов, що отримані на етапі об'єднання, і в процесі накопичення результату дії правил шляхом формування максимуму визначаємо нечітку множину висновку $\{\mu(p)/p\}$, $p \in [0, 10]$.

Числове значення пріоритету p переведення мережі на напругу 20 кВ визначається за методом центра ваги (centroid, COG)

$$p = \frac{\int_{\min}^{\max} p \cdot \mu(p) \cdot dp}{\int_{\min}^{\max} \mu(p) \cdot dp} \quad (4.4)$$

Табл. 4.1. Початкова база знань НЛК

№ правила	Вхідні змінні									Вихідна змінна	Вага правила
	line_6kV	time_net	length_net	load	tengency	densityMW/km2	SAIDI	SAIFI	populat.	priority	
	№ mf	№ mf	№ mf	№ mf	№ mf	№ mf	№ mf	№ mf	№ mf	№ mf	
1	2	0	0	0	0	0	0	0	0	8	1
2	1	0	0	0	0	0	0	0	0	5	1
3	0	1	0	0	0	0	0	0	0	2	1
4	0	2	0	0	0	0	0	0	0	6	1
5	0	3	0	0	0	0	0	0	0	8	1
6	0	0	1	0	0	0	0	0	0	4	1
7	0	0	2	0	0	0	0	0	0	5	1
8	0	0	3	0	0	0	0	0	0	6	1
9	0	0	0	1	0	0	0	0	0	3	1
10	0	0	0	2	0	0	0	0	0	6	1
11	0	0	0	3	0	0	0	0	0	6	1
12	0	0	0	0	1	0	0	0	0	1	1
13	0	0	0	0	2	0	0	0	0	2	1
14	0	0	0	0	3	0	0	0	0	5	1
15	0	0	0	0	0	1	0	0	0	3	1
16	0	0	0	0	0	2	0	0	0	7	1
17	0	0	0	0	0	0	1	0	0	2	1
18	0	0	0	0	0	0	2	0	0	3	1
19	0	0	0	0	0	0	3	0	0	6	1
20	0	0	0	0	0	0	0	1	0	3	1
21	0	0	0	0	0	0	0	2	0	6	1
22	0	0	0	0	0	0	0	0	1	3	1
23	0	0	0	0	0	0	0	0	2	5	1
24	2	3	3	3	3	2	3	2	0	11	1
25	2	2	2	2	2	2	2	2	2	10	1
26	2	2	1	2	2	2	2	2	2	9	1
27	1	1	1	1	2	1	1	2	1	8	1
28	0	2	2	2	0	0	0	0	1	8	1
29	0	0	0	0	0	2	3	2	0	10	1
30	0	3	0	0	0	0	0	0	1	9	1

Для програмного обчислення в дискретному вигляді вираз можна представити у вигляді

$$p = \frac{\sum_{p_{\min}}^{p_{\max}} (p_i + \Delta p) \cdot \mu(p_i + \Delta p) \cdot \Delta p}{\sum_{p_{\min}}^{p_{\max}} \mu(p_i + \Delta p) \cdot \Delta p} . \quad (4.5)$$

Табл. 4.2. Етапи логічного рішення та відповідні оператори

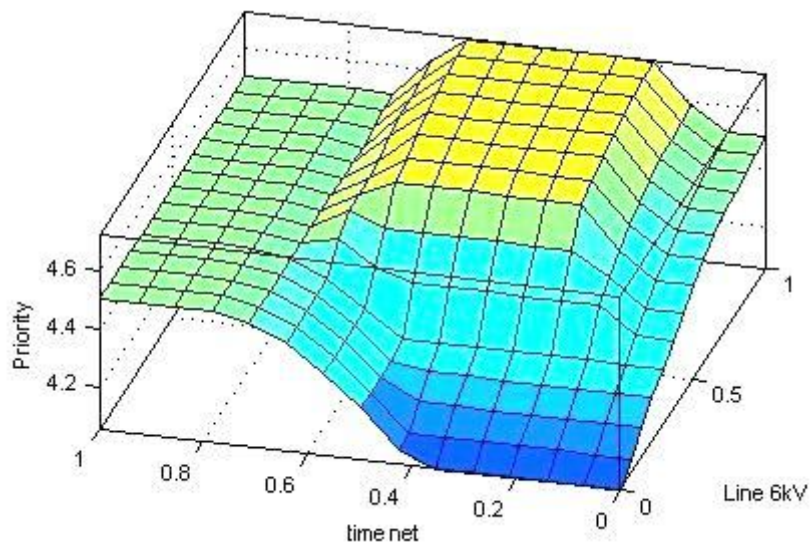
Етап логічного рішення	Оператор
1. Об'єднання вхідних змінних в умовах правил	
оператор «I»	Мінімум
2. Активація висновку правил	
Перетворення «ЯКЩО – ТО висновок $\mu(p)$ »	Мінімум
3. Накопичення результату активованих правил	Максимум

Програмні налаштування НЛК для реалізації нечіткого логічного висновку формуються таким чином:

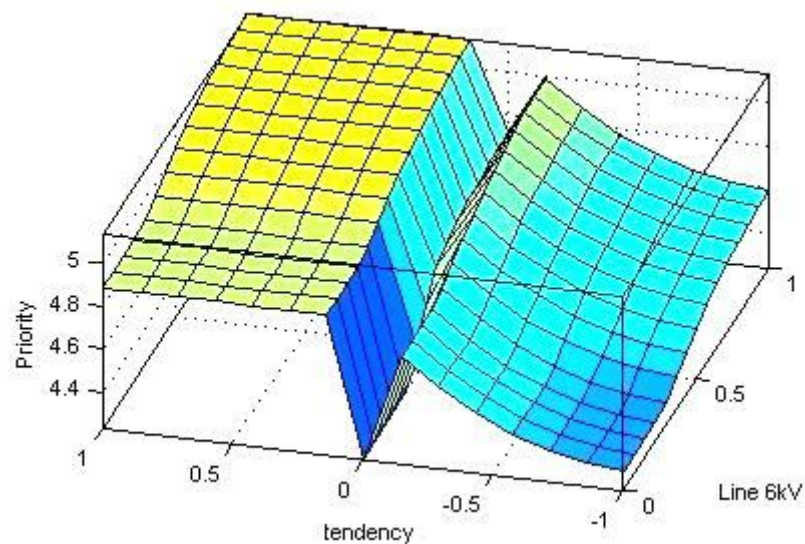
- *Name='prior1';*
- *Type='mamdani' Version=2.0;*
- *NumInputs=9;*
- *NumOutputs=1;*
- *NumRules=30;*
- *AndMethod='min';*
- *OrMethod='max';*
- *ImpMethod='min';*
- *AggMethod='max';*
- *DefuzzMethod='centroid'.*

На початковому етапі моделювання роботи НЛК отримано передатні функції впливу вхідних параметрів стану мережі на пріоритет її переведення на напругу 20 кВ.

На рис. 4.57 зображено характерні залежності пріоритету *priority* від вхідних параметрів *line_6kv*, *time_net*, *tendency* у вигляді поверхні відгуку нечіткого логічного контролера.



a



б

Рис. 4.57. Характерні критичні залежності пріоритету *priority* від вхідних параметрів:
a – *time_net*, *line_6kv*; *б* – *tendency*, *line_6kv*

Як видно з рис. 4.57 характер зміни пріоритету *priority* не відповідає реальним його залежностям від періоду функціонування мережі (*time_net*), сумарної довжини ліній 6 кВ (*line_6kv*) та характеру зміни навантаження (*tendency*). Зі збільшенням *time_net*, тобто часу функціонування мережі пріоритет однозначно має збільшуватися, оскільки мережа з часом зношується. На поверхні відгуку не має бути локальних екстремумів за збільшення довжини ліній напругою 6 кВ, які маємо на поверхні (рис. 4.57, а). Нульова тенденція *tendency* (*zero*) зміни навантаження (як видно з рис. 4.57,б) викликає локальний мінімум пріоритету, чого реально не має відбуватися.

Для усунення неадекватного впливу на пріоритет вхідних параметрів мережі виконано вісім етапів оптимізації з модифікацією функцій належності та коригуванням існуючих і формуванням додаткових 17 правил роботи контролера. В результаті моделювання прийняття логічних рішень створено нечіткий логічний контролер для визначення числового значення пріоритету переведення розподільної мережі на напругу 20 кВ.

Оптимізовані функції належності лінгвістичних змінних *load*, *tendency*, *population* наведені відповідно на рис. 4.58 – 4.60.

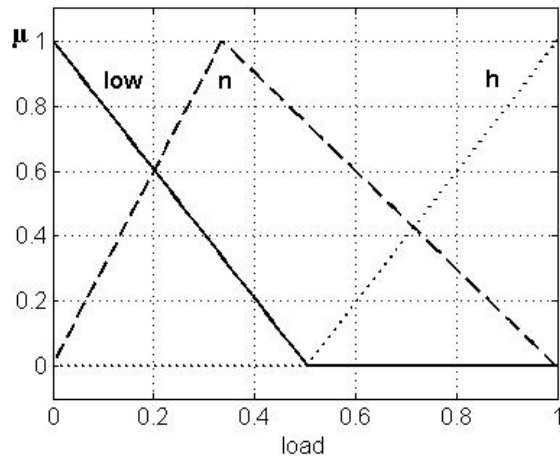


Рис. 4.58. Оптимізовані функції належності лінгвістичної змінної *load*

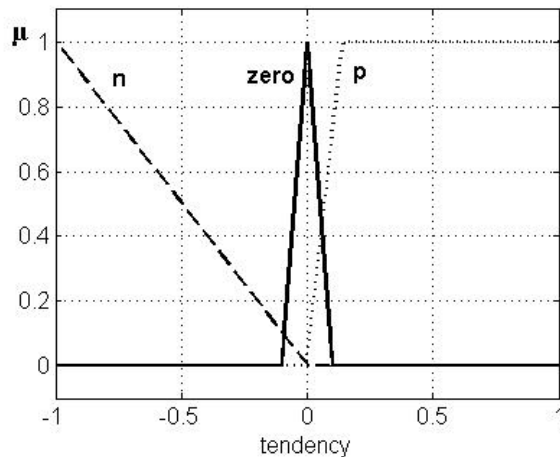


Рис. 4.59. Оптимізовані функції належності лінгвістичної змінної *tendency*

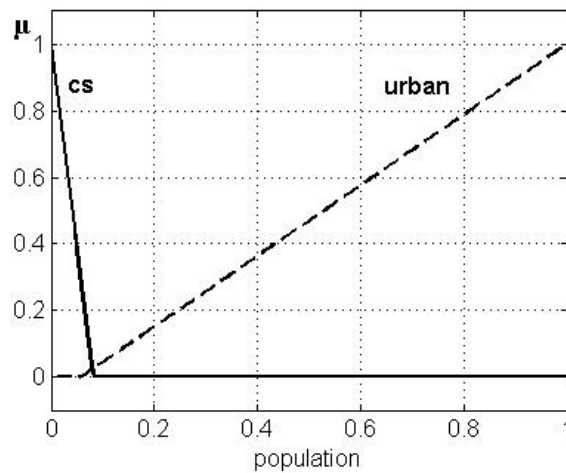


Рис. 4.60. Оптимізовані функції належності лінгвістичної змінної *population*

База знань з 47 правил роботи нечіткого логічного контролера з оптимізованими налаштуваннями наведена в табл. 4.3.

На рис. 4.61 – 4.65 наведені характерні залежності значення пріоритету від параметрів мережі.

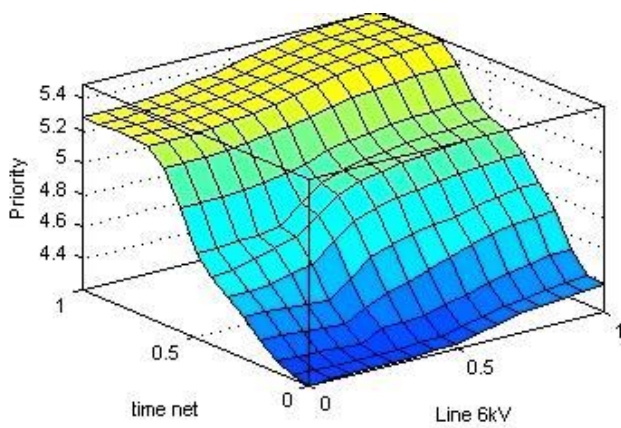
Дослідженням установлено, що при всіх можливих варіаціях значень параметрів, які прийняті для аналізу стану мережі $S = \{line_6kv, time_net, length_net, load, tendency, density, SAIDI, SAIFI, population\}$, у заданих діапазонах їх зміни, значення функції пріоритету переведення розподільної мережі на напругу 20 кВ $P = F(S)$ змінюється в межах від 2,3 до 7,15.

На основі імітаційного моделювання нечіткого логічного контролера у разі зміни в заданих межах вхідних параметрів та аналізу результатів значень пріоритету мережі установлено:

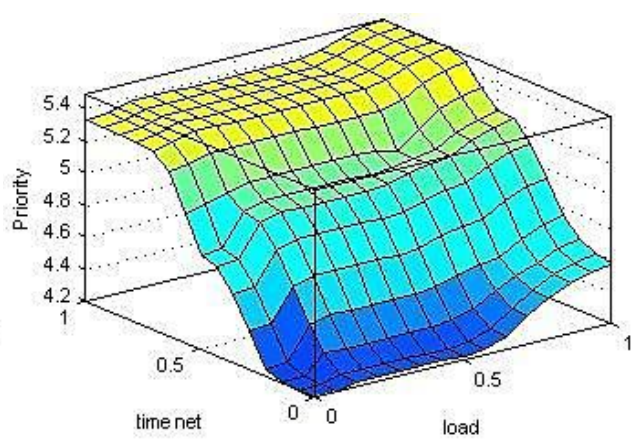
- якщо значення пріоритету переведення мережі на 20 кВ знаходиться в межах від 0 до 2,3, то мережа не підлягає переведенню на 20 кВ;
- якщо значення пріоритету переведення мережі на 20 кВ знаходиться в межах від 2,3 до 4,7, то мережа підлягає переведенню на 20 кВ за умови техніко–економічного обґрунтування;
- якщо значення пріоритету переведення мережі на 20 кВ перевищує значення 4,7, то мережа однозначно підлягає переведенню на напругу 20 кВ.

Табл. 4.3. Оптимізована база знань НЛК

№ правила	Вхідні змінні									Вихідна змінна	Вага правила
	line_6kV	time_net	length_net	load	tengency	densityMW/km2	SAIDI	SAIFI	populat.	priority	
	№ mf	№ mf	№ mf	№ mf	№ mf	№ mf	№ mf	№ mf	№ mf	№ mf	
1	2	3	4	5	6	7	8	9	10	11	12
1	2	0	0	0	0	0	0	0	0	8	1
2	1	0	0	0	0	0	0	0	0	5	1
3	0	1	0	0	0	0	0	0	0	2	1
4	0	2	0	0	0	0	0	0	0	6	1
5	0	3	0	0	0	0	0	0	0	8	1
6	0	0	1	0	0	0	0	0	0	4	1
7	0	0	2	0	0	0	0	0	0	5	1
8	0	0	3	0	0	0	0	0	0	6	1
9	0	0	0	1	0	0	0	0	0	3	1
10	0	0	0	2	0	0	0	0	0	5	1
11	0	0	0	3	0	0	0	0	0	9	1
12	0	0	0	0	1	0	0	0	0	1	1
13	0	0	0	0	2	0	0	0	0	4	1
14	0	0	0	0	3	0	0	0	0	7	1
15	0	0	0	0	0	1	0	0	0	3	1
16	0	0	0	0	0	2	0	0	0	7	1
17	0	0	0	0	0	0	1	0	0	2	1
18	0	0	0	0	0	0	2	0	0	3	1
19	0	0	0	0	0	0	3	0	0	6	1
20	0	0	0	0	0	0	0	1	0	3	1
21	0	0	0	0	0	0	0	2	0	6	1
22	0	0	0	0	0	0	0	0	1	3	1
23	0	0	0	0	0	0	0	0	2	5	1
24	2	3	3	3	3	2	3	2	0	11	1
25	2	2	2	2	2	2	2	2	2	10	1
26	2	2	1	2	2	2	2	2	2	9	1
27	1	1	1	1	2	1	1	2	1	8	1
28	0	2	2	2	0	0	0	0	1	8	1
29	0	0	0	0	0	2	3	2	0	10	1
30	0	3	0	0	0	0	0	0	1	9	1
31	2	3	0	0	0	0	0	0	0	9	1
32	2	1	0	0	0	0	0	0	0	6	1
33	1	3	0	0	0	0	0	0	0	4	1
34	2	0	3	0	0	0	0	0	0	10	1
35	2	0	0	1	0	0	0	0	0	4	1
36	1	0	0	0	2	0	0	0	0	7	1
37	0	1	0	0	2	0	0	0	0	8	1
38	0	0	1	0	2	0	0	0	0	3	1
39	0	0	3	0	2	0	0	0	0	8	1
40	0	0	1	0	3	0	0	0	0	7	1
41	1	0	0	0	0	0	0	0	1	1	0,3
42	0	0	2	3	0	0	0	0	0	8	1
43	0	0	1	1	0	0	0	0	0	2	1
44	0	1	0	0	0	0	1	0	0	1	1
45	0	3	0	0	0	0	1	0	0	10	1
46	0	3	0	0	0	0	2	0	0	10	1
47	1	1	0	0	0	0	0	0	0	1	1

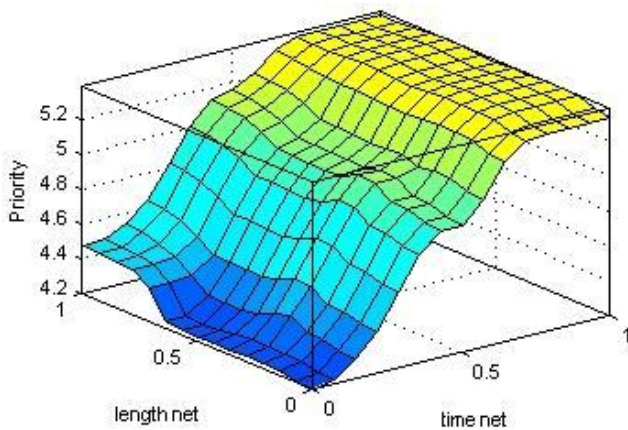


a

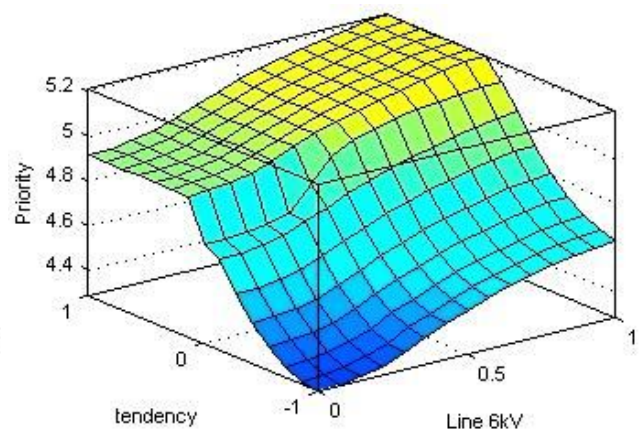


б

Рис. 4.61. Поверхні залежності пріоритету *priority* від вхідних параметрів: *a* – *time_net*, *line_6kv*; *б* – *time_net*, *load*

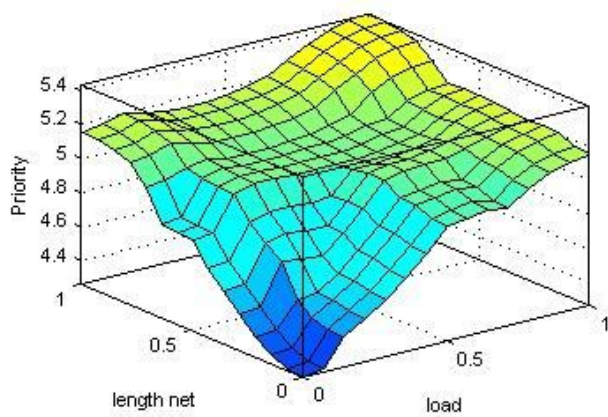


a

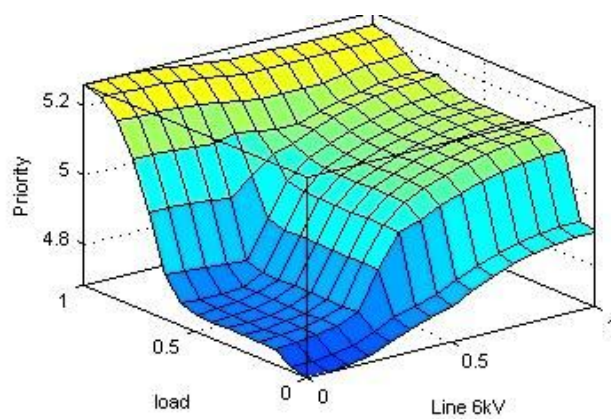


б

Рис. 4.62. Поверхні залежності пріоритету *priority* від вхідних параметрів: *a* – *length_net*, *time_net*; *б* – *tendency*, *line_6kv*

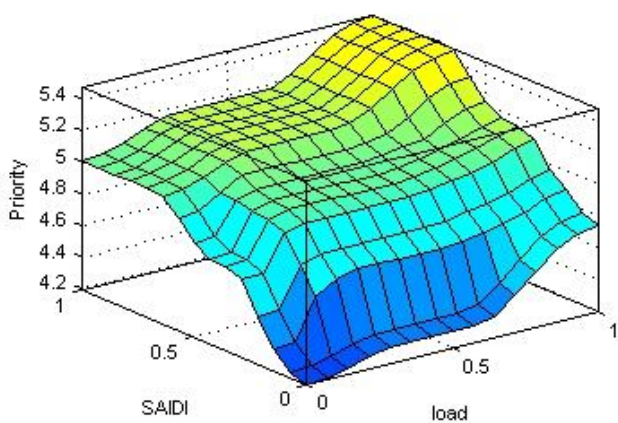


a

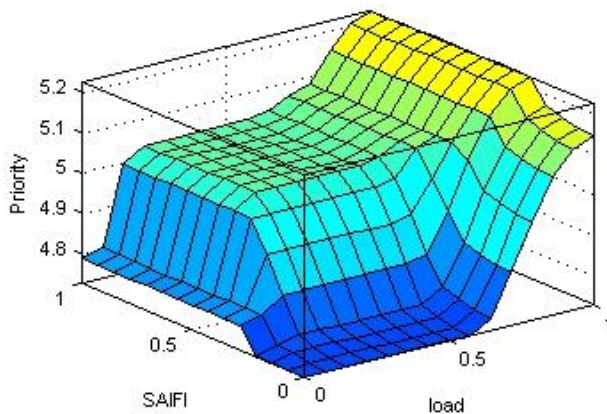


б

Рис. 4.63. Поверхні залежності пріоритету *priority* від вхідних параметрів: *a* – *length_net*, *load*; *б* – *load*, *line_6kv*

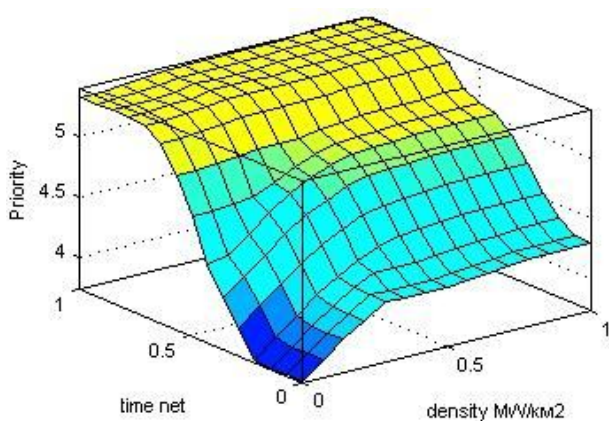


a

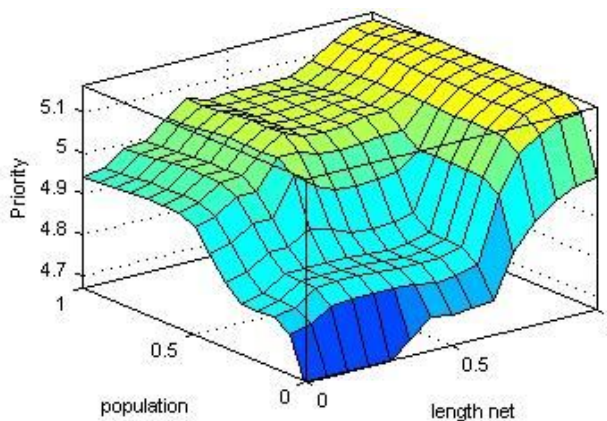


б

Рис. 4.64. Поверхні залежності пріоритету *priority* від вхідних параметрів: *a* – *SAIDI*, *load*; *б* – *SAIFI*, *load*



a



б

Рис. 4.65. Поверхні залежності пріоритету *priority* від вхідних параметрів: *a* – *time_net*, *density_MW/km2*; *б* – *population*, *length_net*

У випадках варіацій параметрів мережі з прийнятими максимальними значеннями діапазонів в абсолютних величинах: *line_6kv* – 50 км; *time_net* – 50 років; *length_net* – 30 км; *load* – 0,8P_{уст}; *tendency* – 0,03P_{уст}/рік; *density_MW/km²* – 10 МВт/км²; *SAIDI* – 300; *SAIFI* – 6; *population* – 500 тис.чол. встановлено, що мережа однозначно підлягає переведенню на напругу 20 кВ за перевищення одним з параметрів таких значень:

- сумарна довжина ліній 6 кВ – 30 км;
- період функціонування – 35 років;
- середня довжина фідерів мережі – 24 км;
- навантаження – 0,8P_{уст} (80 % від установленної потужності);
- територіальна щільність навантаження – 5 МВт/км²;
- населення району електропостачання *population* – 200 тис.чол.

У разі одночасного перевищення значень таких параметрів:

- період функціонування – 20 років;
- середня довжина фідерів мережі – 10 км;
- навантаження – 0,5P_{уст} (50 % від установленної потужності);
- територіальна щільність навантаження – 2 МВт/км²;
- індекс середньої тривалості вимикань *SAIDI* – 60;
- індекс середньої частоти вимикань *SAIFI* – 1,2;
- населення району електропостачання – 50 тис.чол.,

розподільна мережа однозначно має бути переведена на напругу 20 кВ.

КОНТРОЛЬНІ ЗАПИТАННЯ ТА ЗАВДАННЯ

1. Чим викликана необхідність використання нечіткої логіки.
2. Наведіть приклади використання принципів управління з використанням нечіткої логіки.
3. Дайте визначення штучному інтелекту.
4. Дайте визначення нечіткої множини.
5. Яким чином представляється елемент нечіткої множини?
6. Особливості структурної схеми моделі системи з НЛК.
7. Особливості нечіткої моделі з додатковим НЛК.
8. Що таке регулятор з нечітким пропорційно–інтегральним контролером?
9. Чим викликана необхідність створення нечіткого пропорційно – інтегрально–диференціального контролера.
10. Наведіть можливі структурні схеми об'єднання НЛК з класичними регуляторами.
11. Дайте визначення поняттю нечітка логіка.
12. Дайте визначення поняттю лінгвістична змінна.
13. Що собою представляє нечітка змінна?
14. Що таке терм–множина лінгвістичної змінної.

15. Дайте визначення стохастичній невизначеності.
16. Дайте визначення лінгвістичній невизначеності.
17. Обґрунтуйте відмінність нечіткої логіки від теорії ймовірності.
18. Що моделює математичний апарат нечіткої логіки?
19. Дайте визначення нечіткої множини.
20. Яким чином математично визначається нечітка множина?
21. Що таке носії множини?
22. Дайте визначення ядру нечіткої множини.
23. Дайте визначення межах нечіткої множини.
24. Яких числових значень може набувати ступінь функції належності?
25. Яким чином визначається нечітка множина на просторі фізичного параметра?
26. Приведіть основні графічні форми функції належності.
27. Дайте визначення кусково-лінійних функцій належності.
28. В яких випадках використовуються Z-подібні та S-подібні функції належності?
29. В яких випадках використовуються трикутна та трапецієподібна функції належності?
30. Наведіть основні математичні операції над нечіткими множинами.
31. Наведіть графічну інтерпретацію основних математичних операцій над нечіткими множинами.
32. Дайте визначення добутку (перетину) нечітких множин.
33. Дайте визначення суми (об'єднання) нечітких множин.
34. Представлення нечіткого висловлювання (виразу).
35. Дайте визначення логічній кон'юнкції нечітких виразів.
36. Дайте визначення логічній диз'юнкції нечітких виразів.
37. Характеристика нечіткої імплікації, її визначення та суть.
38. Характеристика правил нечітких продукцій (система нечітких правил) та бази знань.
39. Що таке прямий та зворотний методи нечіткого висновку?
40. Базова архітектура систем нечіткого висновку.
41. Дайте характеристику основних етапів висновку.
42. Наведіть основні етапи логічного висновку.
43. Дайте визначення процесу фаззифікації.
44. Дайте визначення процесу агрегації.
45. Дайте визначення процесу активації.
46. Дайте визначення процесу акумуляції.
47. Математичне визначення міні-активації та її графічна інтерпретація.
48. Що представляє собою процес дефаззифікації?

49. *Дайте визначення методу центра ваги.*
50. *Дайте визначення методу центра площі.*
51. *Дайте визначення методу лівого модального значення.*
52. *Дайте визначення методу правого модального значення.*
53. *Дайте визначення алгоритму нечіткого логічного висновку.*
54. *Характеристика алгоритму Мамдані.*
55. *Дайте визначення алгоритму Цукамото.*
56. *Характеристика алгоритму Сугено.*

ЛІТЕРАТУРА

[9], стор. 14–73; [12], стор. 11–15, стор. 24–42, стор. 52–66, стор. 70–98, стор. 134–177, стор. 185–201; [5], стор. 10–27; [8], стор. 9–16, стор. 32–41, стор. 52–64, стор. 126–139; [3], стор. 11–26; [22], стор. 97–108, стор. 232–251; [2], стор. 150–170; [10], стор. 144–150; [1], стор. 3–20; [20], стор. 6–51; [6], стор. 307–310.

РОЗДІЛ 5

ПРАКТИЧНА РЕАЛІЗАЦІЯ СИСТЕМИ З ЕЛЕМЕНТАМИ ШТУЧНОГО ІНТЕЛЕКТУ НА ОСНОВІ НЕЧІТКИХ ЛОГІЧНИХ КОНТРОЛЕРІВ

5.1 Загальні положення та особливості формування параметрів операцій за алгоритмом нечіткого контролера

Нечіткий алгоритм використовується у випадку, коли система може розглядатися як «чорний ящик» з входами, що використовуються для вимірювання, та можливістю їх зміни, тобто коли за об'єктом можна спостерігати та керувати ним в умовах невизначеності та непрогнозованості вхідних параметрів і відсутності формалізованого математичного опису системи. У цьому разі необхідно володіти деякою інформацією щодо функціонування системи або її управління. Вона може бути кількісної або якісної природи і зведена у множину правил (можливо, після деякої обробки).

Прийнятне рішення нечіткого контролера має задовольняти специфікацію розробки (проекту). Воно не має бути оптимальним у відношенні деяких класичних критеріїв, але імітаційна модель системи має візуально довести, що нечітка система керування оптимальна та навіть стійка. Пристрій управління з нечітким алгоритмом у більшості випадків забезпечує стійкість та прийнятне рішення управління установкою.

Система управління з нечітким алгоритмом не повинна використовуватися у таких випадках коли:

- традиційний контролер забезпечує задовільне керування;
- легко розв'язувана та адекватна математична модель уже існує;
- проблема керування не може бути вирішена.

Процедуру виконання алгоритму нечіткого логічного контролера можна представити послідовністю операцій основними з яких є:

- вибір змінних;
- вимірювання змінних;
- масштабування змінних;
- фаззифікація;
- нечіткий висновок (міркування);
- дефаззифікація;
- масштабування вихідного сигналу контролера;
- управління об'єктом (виконавчим пристроєм).

На рис. 5.1 представлено алгоритм нечіткого логічного рішення.

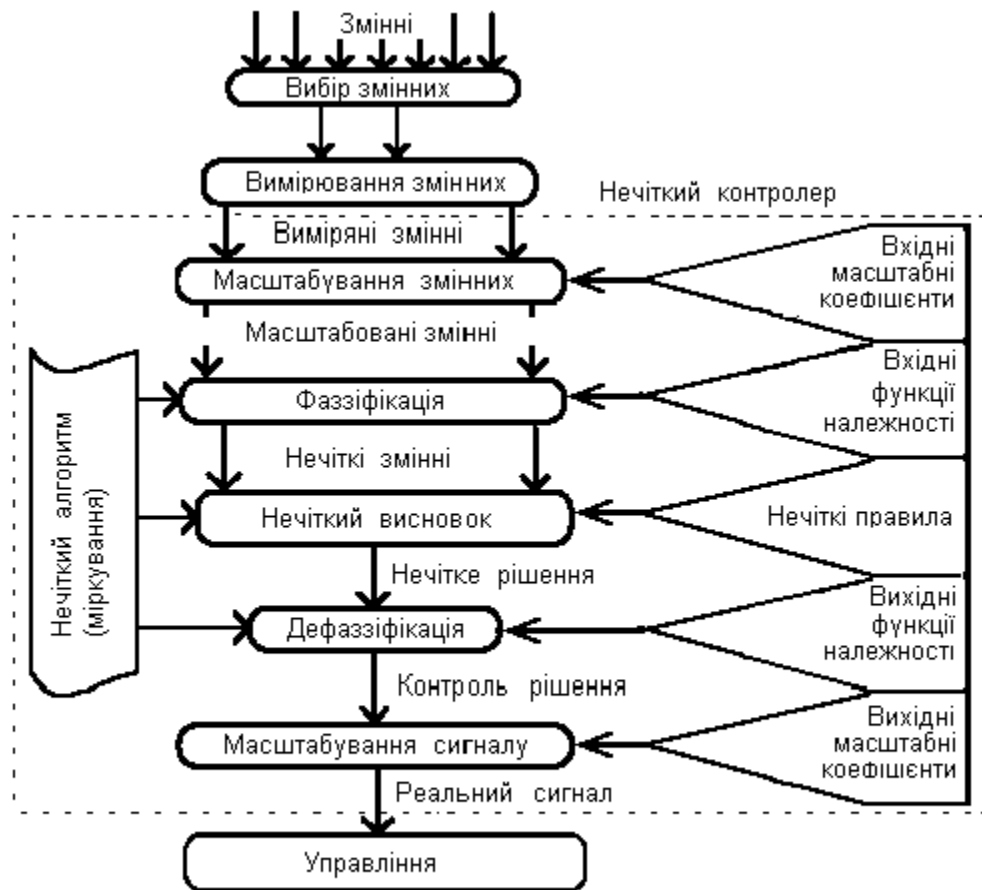


Рис. 5.1. Блок-схема операцій формування рішення щодо управління

Розглянемо особливості структурних елементів нечіткого алгоритму та вибір їх параметрів.

Вибір змінних

Підхід до вибору змінних такий же, що й для звичайної системи управління. Розузгодження та швидкість зміни розузгодження (похідна) сигналів часто використовуються як змінні для пристрою керування з нечітким алгоритмом.

Зауваження

Необхідно вибирати змінні відповідно до технічного або технологічного алгоритму управління, відтворюючи характерну залежність виходу від цих змінних

Вибір масштабних коефіцієнтів

На початку вибираються масштабні коефіцієнти відповідно до прийнятого діапазону дослідження (імітаційні амплітуди) для змінних (якщо відомо характерний діапазон для кожної змінної в контексті роботи системи). Змінюйте масштабні коефіцієнти, щоб забезпечити ефективність параметрів,

тому що розумний вибір коефіцієнтів у одному випадку може стати некоректним у іншому. Пріоритетними є вихідні коефіцієнти.

Зауваження

Бажано, і часто необхідно, масштабувати, або нормалізувати область дослідження змінних, застосовуючи для цього стандартний діапазон $[-1, +1]$.

Вибір функцій належності та їх кількості

Необхідно пам'ятати, що кількість ФН та їхнє позиціонування визначають стратегію поведінки системи (статичну передатну характеристику). Кількість функцій належності (потужність множини лінгвістичної змінної) бажано формувати непарним числом і у першому наближенні вибирати в діапазоні від 3 до 9. Як правило, чим точніше необхідно забезпечити управління (тобто чим більш чутливим має бути вихід до вхідних змінних), тим більшою має бути щільність розміщення функцій належності на вхідній області.

Вибір ФН виконується виходячи із загальної концепції функціонування системи на основі визначення ряду параметрів, що описують лінгвістичну змінну на фізичному діапазоні (базовій змінній або фізичному параметрі):

- типу функції належності;
- форми ФН (нахилу кривої);
- позиціонування ФН на області дослідження;
- ширини функцій належності (діапазон носія нечіткої множини).

Форма кривих ФН, їхні зони перекриття та точки перетину істотно впливають на коефіцієнт передачі та чутливість контролера.

Загальні рекомендації:

- у разі невідомої передатної характеристики системи вибирають ширину носія функцій належності приблизно 12...14 % ширини діапазону лінгвістичної змінної, щоб забезпечити повне його покриття з однаковим перекриттям і з перетином ФН на рівні 0,5 ступеня належності.
- у разі відомої передатної характеристики системи можливий вибір ширини ФН за половиною значення ділянки між характерними точками або локальними екстремумами на характеристиці, а також вибір ширини носія кожної ФН за відповідними проекціями лінійних ділянок, що апроксимують криву передатної характеристики;
- для зменшення усталеної похибки та часу формування рішення управління необхідно зменшувати зони перекриття ФН;
- для поліпшення характеристики перехідного процесу (коливання, час затухання, перерегулювання) необхідно збільшувати зони перекриття ФН;
- використання пристрою управління з нечітким алгоритмом з більш широким носієм ФН і більшим перекриттям можна рекомендувати за наявності значних збурень. Використання вузьких носіїв ФН призводить

систему до зниження усталеної похибки, але у разі дуже вузького носія ФН імовірність досягнення усталеного стану системи занадто низька.

Вибір правил

Формування правил базується на таких аспектах:

- досвід експерта та знання;
- специфікація управління для оператора;
- імітаційна модель процесу або об'єкта управління, технічне завдання;
- вивчення методик управління аналогічними об'єктами.

Набір правил бази знань має бути: повним; несуперечливим; неперервним.

Вибір методу дефаззифікації

Вибір методу необхідно проводити згідно зі стандартом Міжнародної Електротехнічної Комісії ІЕС 1131–7.

Зауваження

Найбільш широко для електроенергетичних задач використовуються методи центра ваги та центра площі.

Вибір алгоритму нечіткого логічного висновку (міркування)

Алгоритм Mamdani використовують у випадку складної нелінійної системи та неможливості математичного опису системи, коли правила будуть сформульовані експертом.

Алгоритм Takagi–Sugeno використовують, коли існує досить достовірний математичний опис (можливо, фрагментарний) і коли важливо досягти необхідної обчислювальної ефективності (ККД) і зручності аналізу перехідних процесів (стійкості системи за Ляпуновим).

Приклад 5.1

Синтез НЛК для визначення ступеня достовірності враження блискавкою опор високовольтних ліній електропередавання на основі нечіткого логічного висновку для оптимального розміщення нелінійних обмежувачів перенапруги (ОПН).

Щільність розрядів блискавки на землю залежно від місцевості проходження ЛЕП (грунти, рельєф, наявність водойм та водоносних пластів неглибокого залягання, ширини грозового фронту) за однакового значення кількості грозових годин може відрізнятися і досить суттєво. Тому розрахункове число ударів блискавки в лінію на основі існуючої методики¹ є досить наближеним. Розподілення ударів блискавки за прогоном не можливо достовірно математично формалізувати, тому точність визначення числа можливого перекриття ізоляції є досить проблематичним моментом у таких умовах.

¹ Руководство по защите электрических сетей 6 – 1150 кВ от грозовых и внутренних перенапряжений РД 153-34.3-35.125-99 / Под науч. ред.–2-е издание. СПб.: ПЭИПК Минтопэнерго РФ, 1999.

Виходячи з існуючої методики, очікуване число грозових вимкнень від зворотних перекриттів лінійної ізоляції у разі удару блискавки в опору визначається через кількість ударів блискавки в опору N_{on} , імовірність перекриття лінійної ізоляції P_{on} та коефіцієнт переходу імпульсного перекриття в дугу промислової частоти η таким чином:

$$n_{on} = N_{on} P_{on} \eta .$$

Кількість ударів блискавки в опору визначається через загальну кількість N дарів блискавки в ЛЕП на 100 км помножену на чотири відношення середньої висоти підвісу троса на опорі $h_{тр}$ до довжини прогону $l_{прог}$ за виразом

$$N_{on} = 4N \frac{h_{тр}}{l_{прог}} .$$

Аналіз виразу показує, що кількість ударів блискавки в опору залежить тільки від типу опор та розрахункового середнього прогону $l_{прог} = 0,89l_{габ}$, але не враховує реальної довжини прилеглих до кожної з опор прогонів та зміни висоти рельєфу, які суттєво можуть впливати на кількості ударів блискавки в опору.

Для синтезу НЛК у попередній наведеній формулі чотири відношення середньої висоти підвісу троса на опорі $h_{тр}$ до довжини прогону $l_{прог}$ представимо ступенем достовірності враження $\mu_i(N)$ i -ої опори блискавкою, тобто

$$N_{on} = \mu_i \cdot N .$$

Ступінь достовірності враження є нечіткою логічною функцією від суміжних лівого та правого прогонів (відносно опори) і висоти точки підвісу троса на опорі

$$\mu_i(N) = f(LL, LR, HS) ,$$

де LL – лінгвістична змінна довжини $L1$ прогону до опори, що розглядається; LR – лінгвістична змінна довжини $L2$ прогону після опори, тобто наступного прогону по трасі ЛЕП; HS – лінгвістична змінна висоти h , відстані від точки підвісу троса на опорі до точки над опорою на умовній прямій лінії, що з'єднує точки підвісу троса на сусідніх опорах (рис.5.2); i – номер опори.

Для реалізації функції проведено синтез нечіткого логічного контролера в пакеті Fuzzy програмного середовища Matlab. Дві вхідні лінгвістичні змінні з термами LL та LR (довжин прогонів) представлено аналогічно п'ятьма нечіткими змінними (рис. 5.3), які описано трикутними функціями належності: B – великий, BA – більше середнього, A – середній, LA – менше середнього, S – малий. Третя вхідна лінгвістична змінна HS (висота підвішування троса) представлена трьома нечіткими змінними (рис. 5.4), які також описано трикутними функціями належності: P – позитивна, Z – нульова, N – негативна.

Вихідна лінгвістична змінна μ_i ступінь достовірності враження блискавкою

опори лінії електропередавання, представлена термом PES, описано п'ятьма нечіткими змінними (рис. 5.5), які, як і вхідні, представлені трикутними функціями належності: Н – висока, ВА – вище середньої, А – середня, LA – нижча середньої, N–низька.

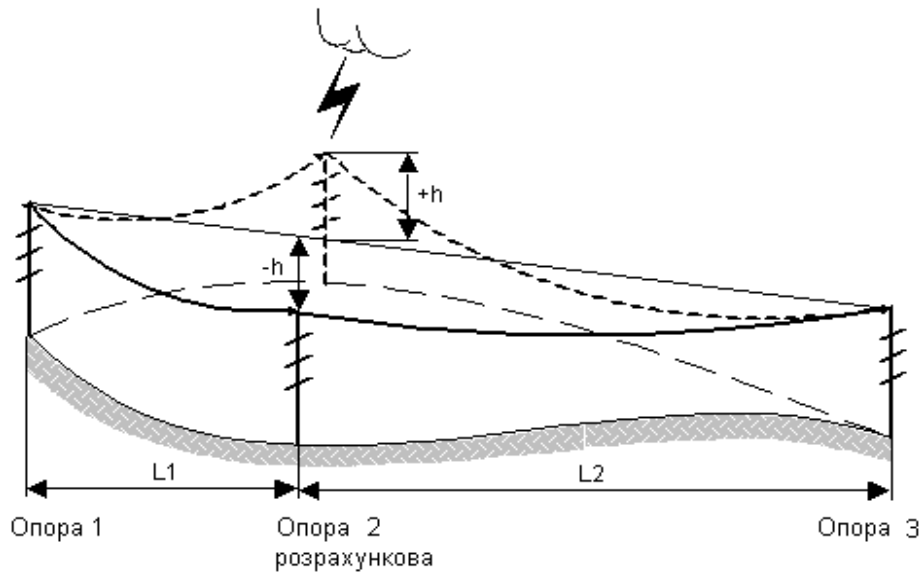


Рис. 5.2. Розрахункова схема профілю траси

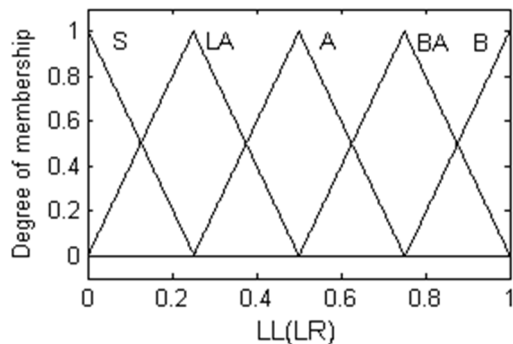


Рис. 5.3. Лінгвістичні змінні LL (LR)

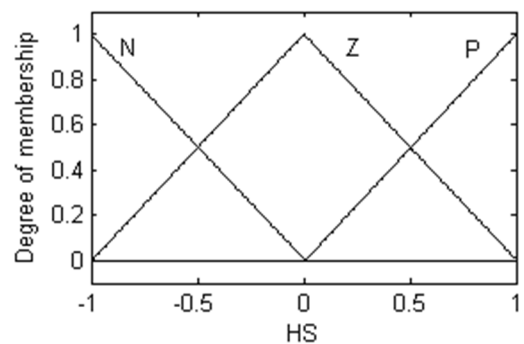


Рис. 5.4. Лінгвістична змінна HS

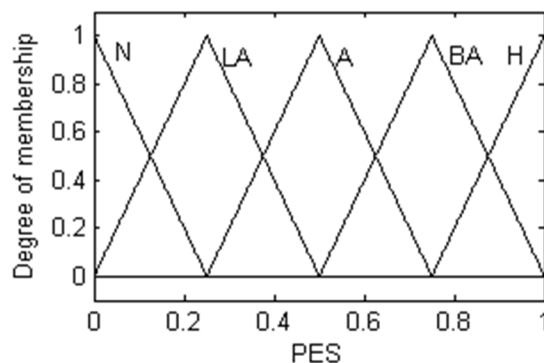


Рис.5.5. Вихідна лінгвістична змінна PES

Для формування логічного висновку сформована база знань нечіткого логічного контролера з 75 правил формату:

1. Якщо $(LL \in S) \text{ і } (LR \in S) \text{ і } (HS \in P)$, то $(PES \in BA)$ (1);
-;
75. Якщо $(LL \in B) \text{ і } (LR \in B) \text{ і } (HS \in N)$, то $(PES \in N)$ (1).

Усі правила з максимальним ваговим коефіцієнтом, що дорівнює одиниці, представлені в табл. 5.1 можливими варіаціями трьох входних лінгвістичних змінних, дві з яких мають по п'ять, а одна три нечіткі змінні.

Табл. 5.1. База знань нечіткого логічного контролера
(незатінені терми вихідної змінної PES)

HS		N					Z					P				
LR		S	LA	A	BA	B	S	LA	A	BA	B	S	LA	A	BA	B
LL	S	BA	A	LA	LA	LA	BA	BA	LA	LA	LA	BA	BA	A	LA	LA
	LA	A	LA	LA	LA	N	BA	A	LA	LA	LA	BA	A	A	LA	LA
	A	LA	LA	LA	N	N	LA	LA	LA	LA	N	LA	A	LA	LA	LA
	BA	LA	LA	N	N	N	LA	LA	LA	N	N	LA	LA	LA	LA	N
	B	LA	N	N	N	N	LA	LA	N	N	N	LA	LA	LA	N	N

Після моделювання роботи контролера проведено коригування бази знань – добавлені три правила, які мають такий вигляд:

76. Якщо $HS \in P$, тоді $PES \in BA$ (0,3);
77. Якщо $HS \in Z$, тоді $PES \in A$ (0,3);
78. Якщо $HS \in N$, тоді $PES \in LA$ (1).

Структурну схему синтезованого нечіткого логічного контролера показано на рис.5.6.

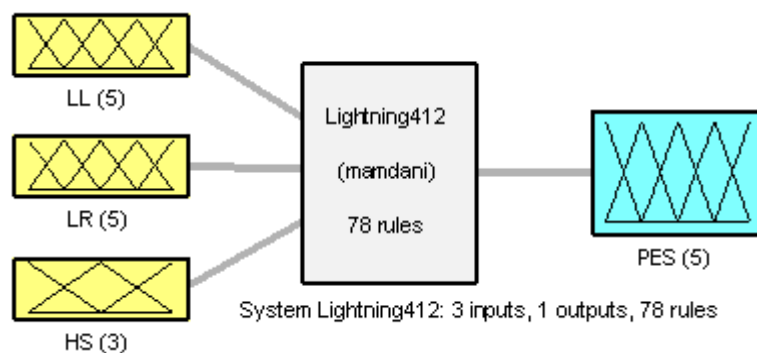


Рис. 5.6. Структурна схема НЛК

На рис. 5.7 наведено поверхню відгуку нечіткого логічного контролера.

Шляхом послідовного моделювання в НЛК (на основі вхідних параметрів для кожної з опор по трасі ЛЕП) знаходимо значення вихідних змінних PES, тобто ступінь достовірності μ_i враження блискавкою кожної з опор траси, де $i = 2, \dots, (n-1)$, за виключенням крайніх, якими виконано заходи до підстанцій. Після цього розраховується середнє значення ступеня достовірності для опор всієї траси:

$$\mu_{cp}(N) = \frac{\sum_{i=2}^{n-1} \mu_i(N)}{n-2}.$$

Виходячи з почергового порівняння ступенів достовірності враження блискавкою кожної з опор з середнім значенням, визначається кількість опор j та ступені достовірності їх враження блискавкою $\mu_{i,j}(N)$, які перевищують середнє значення, і виходячи з умови техніко-економічного обґрунтування кількості опор m , на яких необхідно установити обмежувачі перенапруги, знаходиться m максимальних значень $\mu_{i,j}(N)$ і відповідно номери їх опор, на яких рекомендована установка ОПН, згідно з виразом

$$\mu_{i,m}(N) = \max_{m \in j} \left\{ \max_{\forall i \in \{2, \dots, (n-1)\}} \{ \mu_i(N) - \mu_{cp}(N), 0 \} \right\},$$

де n – кількість опор лінії електропередавання.

Використання математичного апарата нечіткої логіки дає змогу на етапі проектування лінії електропередавання визначити номери опор з максимальними значеннями ступеня достовірності враження їх блискавкою та відповідно до цього сформулювати схему розміщення обмежувачів перенапруги вздовж траси ЛЕП на основі міркувань нечіткої логіки, виходячи з бази знань, отриманої шляхом імітаційного моделювання.

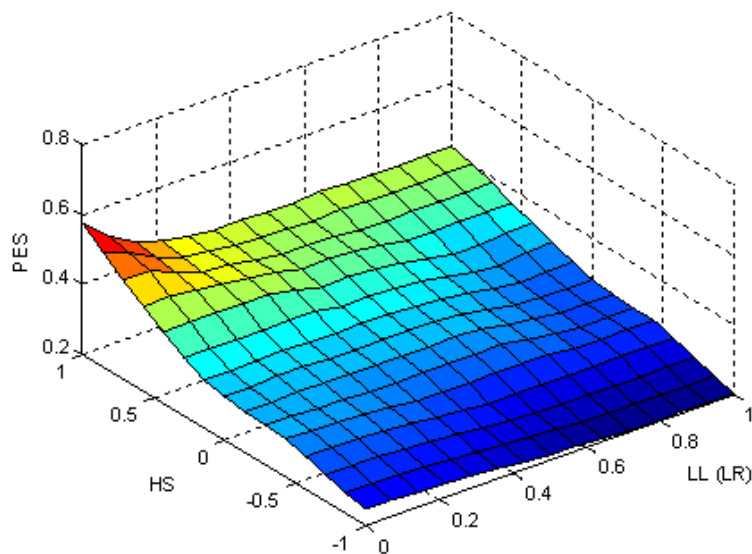


Рис. 5.7. Поверхня відгуку НЛК

5.2 Формування функцій належності, правил та налаштування нечіткого логічного контролера

У табл. 5.2 наведено узагальнені підходи, методи та джерела прикладної інформації, а також короткий опис процедури формування функцій належності.

Табл. 5.2. Джерела прикладної інформації

Підхід	Метод	Джерело прикладної інформації	Короткий опис процедури
1	2	3	4
Суб'єктивний	Професійні знання експерта та інтуїція	Основає на здатності експерта генерувати інформацію через знання та розуміння проблеми та (або) прикладної області	Експерт визначає ступінь належності для будь-якого числового значення діапазону або пропонує функцію належності для визначення нечіткої множини. Нечітка змінна наводиться лінгвістичними термінами
	Оброблення результатів опитування думки фахівців	Таке, як попереднє, але стосовно до декількох експертів з узагальненою обробкою	Як попередній, але ступені належності, призначені індивідуально експертами, обробляються в цілому для одержання загальної характеристики
	Ранжування	Основає на здатності експерта (ів) зрівнювати і установлювати пріоритет для різних об'єктів або числових значень	Експерт(и), увівши нормовані (характерні) значення параметра об'єкта та оцінюючи ранги значень на діапазоні, шляхом попарного їхнього порівняння установлюють ступені належності
	Логічний висновок	Основає на висновках, виходячи з наявних знань (закони природи, практичні знання і т.п.)	Ступені належності визначаються з деякої інформації, що є в наявності, і пов'язуються з розглянутим об'єктом
	Індуктивне міркування	Основає на одержанні ступеня належності від часткових фактів (набору даних)	Ступені належності одержують узагальненням деяких наявних фрагментарних даних

Табл. 5.2. (продовження)

1	2	3	4
Об'єктивний	Нечітка статистика та (або) кластеризація	Основа на статистичній обробці отриманих даних	Ступені належності отримують математичними методами статистики, теорії кластеризації нечітких центрів, кластеризації віднімання та теорії нечіткої міри
	Розробка систем управління	Основа на формуванні функцій належності на основі рекомендацій теорії управління	Функції належності сформовані відповідно до деяких правил, отриманих на основі класичних методів теорії управління
	Нейронні мережі	Основа на моделюванні та визначенні функцій належності і їх параметрів у нейронній мережі	У процесі моделювання нейронної системи нейронною мережею формуються та (або) коригуються функції належності
	Генетичні алгоритми	Основа на визначенні функцій належності шляхом генетичних (еволюційних) алгоритмів	Прийняті параметри функцій належності змінюються шляхом спеціальної методики оптимізації на основі схрещування, мутації та селекції

На рис. 5.8 схематично наведено основні джерела та шляхи формулювання правил.

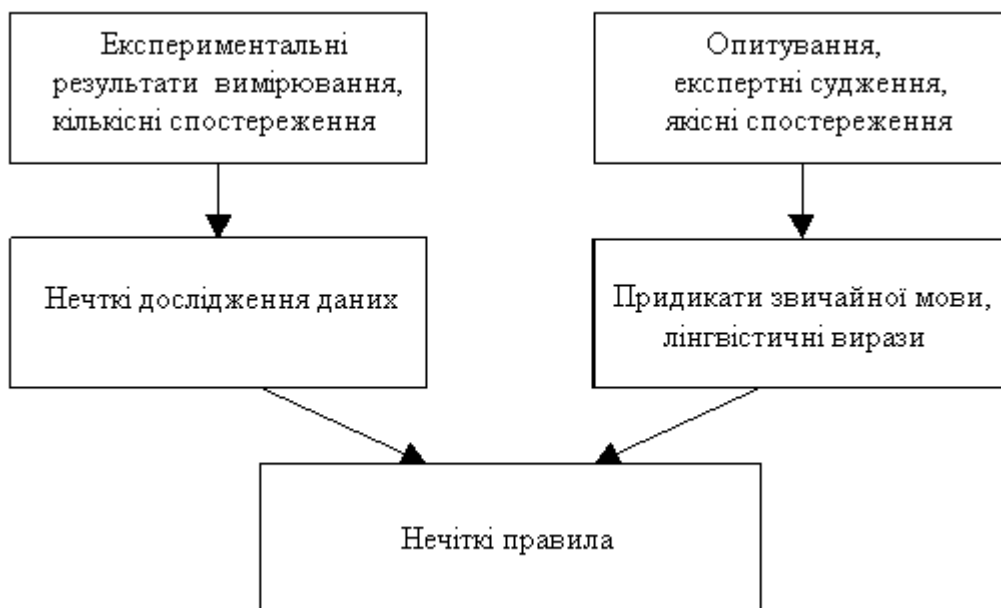


Рис. 5.8. Схема формулювання правил

Для перевірки та налаштування НЛК найбільш широко використовується моделювання, оскільки на сьогодні не існує достовірної теорії аналізу пристрою управління з нечітким алгоритмом, бо досить складно теоретично описати прийняття того або іншого рішення управління на основі змодельованих розумових міркувань. Основні та найбільш важливі критерії, яким пристрій керування з нечітким алгоритмом має задовольняти – це здоровий глузд, професійне чуття та знання експертом прикладної області, а також алгоритму (програми) формування нечіткого рішення.

До найбільш важливих критеріїв можна віднести:

1. Пристрій управління з нечітким алгоритмом налаштовується разом з реальним об'єктом або ж з математичною моделлю, якщо вони є в наявності;
2. Необхідно використовувати один (спільний) прикладний програмний пакет для моделювання пристрою керування з нечітким алгоритмом та об'єкта управління;
3. У процесі моделювання необхідно використовувати різні комбінації вхідних змінних та проводити візуальне спостереження відповідних виходів та поверхні відгуку;
4. Значення змінних, які можуть викликати порушення роботи пристрою керування з нечітким алгоритмом:
 - вхідні значення у критичних зонах області дослідження;
 - вхідні значення у критичній (характерній, екстремальній) зонах локальних областей функції належності;
 - вхідні значення в області перекриття функцій належності.

Програмні інструментальні засоби моделювання НЛК мають у своєму складі опції для створення поверхні управління. Аналіз поверхні управління дає змогу контролювати відповідність поверхні управління математичним очікуванням та адекватністю функціонування об'єкта управління заданим критеріям. Відсутність відповідності у цьому контексті звичайно зводиться до однієї з трьох проблем:

- неправильно визначені правила;
- некоректно сформовані функції належності;
- неправильно вибрані нечіткий алгоритм висновку та нечіткі оператори.

Моделюючи роботу НЛК необхідно переконатись, що аномальні або сильні спотворення поверхні (максимальний коефіцієнт передачі вхід–вихід) не сконцентровані на границях для всіх характерних представлень поверхні. Якщо це так, то необхідно заново провести налаштування та перевірити правильність прийнятих висновків у правилах або ввести додаткові правила, використовуючи асоціативне відображення у зонах аномалій, модифікувати масштабні коефіцієнти та (або) вагові коефіцієнти відповідних правил, змінити

щільність позиціювання функцій належності в області максимальних відхилень на поверхні.

Виявлені в процесі моделювання роботи НЛК проблеми функціонування усуваються відповідними діями.

У табл. 5.3 наведено вирішення деяких можливих проблем систем управління з нечітким логічним контролером.

Табл. 5.3. Узагальнені проблеми НЛК та їх вирішення

Проблема	Вирішення
1	2
Пристрій управління з нечітким алгоритмом не забезпечує стабільності системи	1. Змінити структуру системи керування. Введіть додатковий параметр контролю виходу системи в пристрій керування. 2. Перевірити таблицю правил: іноді необхідно змінити неправильний знак або сам висновок у правилі. 3. Змінити метод дефаззифікації або зменшити масштабний коефіцієнт виходу нечіткого контролера
Перерегулювання (величина коливання виходу) занадто висока	1. Зменшити масштабний коефіцієнт виходу або вагові коефіцієнти правил. 2. У PID-подібному пристрої управління з нечітким алгоритмом зменшити коефіцієнти PD-частини
Швидкодія занадто низька (час наростання занадто тривалий)	1. У PID-подібному пристрої управління з нечітким алгоритмом збільшити коефіцієнти PD-частини. 2. Збільшити кількість та(або) змінити форму ФН вихідної змінної. Можливо, змінити висновок у правилах
Недостатня усталена точність	1. У PID-подібному пристрої керування з нечітким алгоритмом зменшити коефіцієнти PD-частини. 2. Збільшити масштабний коефіцієнт для інтегрального входу. 3. Змінити метод дефаззифікації. 4. Зменшити ширину функції належності для незначного сигналу розузгодження. 5. Змістити вершини модальних функцій належності в зоні низьких розузгоджень, збільшуючи їх концентрацію до нульової точки

Табл. 5.3. (продовження)

1	2
Недостатня чутливість до вхідного сигналу	Збільшити масштабний коефіцієнт для цього входу
Аномалії на поверхні управління	1. Перевірити таблицю правил. Вона може містити некоректні правила. 2. Перевірити таблицю правил за заданими властивостями. 3. Перевірити зони перекриття всіх функцій належності
Недостатня швидкодія (ефективність)	1. Змінити метод дефаззифікації та (або) нечіткий алгоритм висновку. 2. Розглянути інше виконання (спеціалізований цифровий процесор або ПЛІС разом з RISC мікроконтролером)

5.3 Синтез та моделювання нечіткого логічного контролера з використанням програмного пакета FUZZY в середовищі MATLAB

Нечітке моделювання є на сьогодні актуальним та перспективним напрямком сучасних інтелектуальних технологій для дослідження складних нелінійних систем та процесів прийняття рішень щодо управління.

Традиційні класичні методи побудови моделей не є ефективними, коли досліджуваний об'єкт чи система не мають повного адекватного формалізованого опису. Сучасна наука має досить потужні математичні засоби для отримання вичерпної інформації у випадку вирішення складних нелінійних задач, але розв'язання ними проблематики може призвести до втрати часу та ресурсів і не може гарантувати в принципі побудову адекватної моделі.

На сьогодні у разі вирішення задач моделювання складних систем в енергетиці найбільш доцільним є використання методів, що орієнтовані на побудову моделей в умовах невизначеності та недостовірності вхідних даних. Технологія нечіткого моделювання саме і побудована на таких методах. Ці методи ґрунтуються на теорії нечітких множин та нечіткої логіки.

Нечітка модель – це інформаційно-логічна модель системи чи підсистеми, що побудована на основі теорії нечітких множин та нечіткої логіки.

Нечітке моделювання є складовою системного моделювання і покликане конкретизувати ті чи інші елементи або вузли моделі системи.

Технологія нечіткого моделювання включає такі етапи:

- Аналіз проблемної ситуації;
- Структуризація предметної області і створення нечіткої моделі;
- Дослідження нечіткої моделі;
- Обробка результатів дослідження;
- Коригування моделі.

Найбільш поширеними програмними пакетами нечіткого моделювання є додаток (toolbox) FUZZY в системі MATLAB та спеціалізоване програмне середовище FUZZYTECH.

5.3.1 Програмний пакет нечіткої логіки FUZZY в середовищі MATLAB та його особливості

Додаток нечіткої логіки (**Fuzzy Logic Toolbox**) до програмного пакета **MATLAB** має простий та добре продуманий інтерфейс, який дає змогу легко проектувати і діагностувати нечіткі моделі. Забезпечуються підтримка сучасних методів нечіткої кластеризації та адаптивні нечіткі нейронні мережі. Графічні засоби дають змогу інтерактивно відслідковувати особливості поведінки системи. Fuzzy Logic Toolbox використовується разом з Simulink. За допомогою Real-Time Workshop можливо генерувати ANSI C код для роботи в реальному часі.

Розглянемо структуру Fuzzy Logic Toolbox. Цей додаток включає такі категорії програмних інструментів:

1. Функції;
2. Інтерактивні модулі з графічним інтерфейсом користувача (GUI);
3. Блоки для пакета Simulink;
4. Демонстраційні приклади;
5. С-код машини нечіткого логічного висновку.

Перша категорія програмних інструментів пакета Fuzzy Logic Toolbox включає функції, які можуть бути викликані безпосередньо шляхом набору імені функції в командному вікні (command line) або з власних доповнень користувача. Більшість з цих функцій є функціями **MATLAB** у вигляді m-файлів. У цьому випадку користувач може проглянути запрограмовані в цих функціях алгоритми, а також редагувати та коригувати ці файли. Нижче наведені назви функцій з коротким описом їх призначення:

1. **addmf** – додати функції належності в FIS (Fuzzy Inference System – система нечіткого висновку);
2. **addrule** – додати правила в FIS;
3. **addvar** – додати змінну в FIS;
4. **anfis** – навчання FIS типу Сугено (Sugeno type);
5. **convertfis** – перетворення FIS-матриці (Fuzzy Logic Toolbox v.1) в FIS-структуру (Fuzzy Logic Toolbox v.2);

6. **defuzz** – дефаззифікація нечіткої множини;
7. **discfis** – дискретизація функцій належності всіх термів, що входять в FIS;
8. **dsigmf** – функція належності у вигляді різниці між двома сигмовидними функціями;
9. **evalfis** – виконання нечіткого логічного висновку;
10. **evalmf** – обчислення значень довільної функції належності;
11. **evalmmf** – розрахунок ступенів належності для декількох функцій належності;
12. **fcm** – пошук кластерів за алгоритмом fuzzy c-means;
13. **findrow** – знаходження рядка в матриці, що співпадає з вхідним рядком;
14. **fstrvc** – конкатенація матриць різного розміру;
15. **fuzarith** – нечіткий калькулятор;
16. **gauss2mf** – двостороння гаусова функція належності;
17. **gaussmf** – гаусова функція належності;
18. **gbellmf** – узагальнена дзвоноподібна функція належності;
19. **genfis1** – генерування з даних вихідної (початкової) FIS типу Сугено без використання кластеризації;
20. **genfis2** – генерування з даних вихідної FIS типу Сугено з використанням субтрактивної кластеризації;
21. **genparam** – генерування початкових параметрів функцій належності для навчання ANFIS (Adaptive–Network–based Fuzzy Inference System);
22. **gensurf** – генерування поверхні «входи–вихід», відповідної FIS;
23. **getfis** – отримання властивостей FIS;
24. **mam2sug** – перетворення FIS типу Мамдані в FIS типу Сугено;
25. **mf2mf** – перерахунок параметрів вбудованих функцій належності різних типів;
26. **newfis** – створення нової FIS;
27. **parsrule** – вставка в FIS правил, що задані у вигляді речень природною мовою;
28. **pimf** – пі-подібна функція належності;
29. **plotfis** – виведення основних параметрів FIS у вигляді графічної схеми;
30. **plotmf** – виведення графіків функцій належності термів однієї змінної;
31. **probor** – імовірнісна реалізація логічної операції «АБО» («ИЛИ»);
32. **psigmf** – добуток двох сигмовидних функцій належності;
33. **readfis** – завантаження FIS із файла;
34. **rmmf** – видалення функції належності терма з FIS;
35. **rmvar** – видалення змінної з FIS;
36. **setfis** – установка властивостей FIS;
37. **showfis** – виведення на екран в текстовому форматі даних, що складають FIS-структуру;
38. **showrule** – виведення бази знань FIS;
39. **sigmf** – сигмоїдна функція належності;

40. **smf** – s-подібна функція належності;
41. **subclust** – оцінка кількості кластерів у субтрактивній кластеризації;
42. **sugmax** – знаходження діапазону зміни вихідної змінної в FIS типу Сугено;
43. **trapmf** – трапецієвидна функція належності;
44. **trimf** – трикутна функція належності;
45. **writefis** – збереження FIS на диску;
46. **zmf** – z-подібна функція належності;
47. **distfcm** – розрахунок відстані за Евклідом;
48. **initfcm** – генерування відправної матриці ступенів належності для нечіткої c-means кластеризації;
49. **isfis** – перевірка структури даних системи нечіткого логічного висновку.

Зауваження

Функції `findrow` та `fstrvcat` не зв'язані з нечіткими множинами та нечіткою логікою – вони використовуються для виконання алгоритмів обробки інформації загального характеру, необхідної для других функцій Fuzzy Logic Toolbox.

Друга категорія програмних інструментів пакета Fuzzy Logic Toolbox включає діалогові модулі, які забезпечують доступ до більшості функцій через графічний інтерфейс. Окрім цього, ці модулі забезпечують зручне середовище для проектування, дослідження та впровадження систем на основі нечіткого логічного висновку. Для запуску інтерактивних модулів достатньо надрукувати ім'я модуля в командному рядку. Нижче наведені назви модулів з коротким описом їх призначення:

1. **anfisedit** – модуль для генерування з даних FIS типу Сугено, його навчання з використанням ANFIS алгоритму та тестування;
2. **findcluster** – модуль кластеризації даних з використанням алгоритму fuzzy c-means та алгоритму нечіткої субтрактивної кластеризації;
3. **fuzzy** – основний редактор FIS. Дає змогу створювати та редагувати FIS двох типів – Мамдані та Сугено, забезпечує візуалізацію процедури нечіткого логічного висновку та поверхні «входи–вихід». Для цього модуль fuzzy викликає такі GUI-модулі: `mfedit`, `ruleedit`, `ruleview` та `surfview`;
4. **mfedit** – редактор функцій належності. Дає змогу вибирати тип функції належності та установлювати її параметри в символьному та інтерактивному графічному (drag) режимах;
5. **ruleedit** – редактор бази знань;
6. **ruleview** – модуль візуалізації процедури нечіткого логічного висновку. Забезпечує виведення графічної діаграми нечіткого висновку за кожним правилом, включаючи процедури фаззифікації, агрегації та дефаззифікації. Дає змогу вводити значення вхідних змінних у символьному та інтерактивному графічному (drag) режимах.

Третя категорія програмних інструментів пакета Fuzzy Logic Toolbox включає такі модулі, які забезпечують інтеграцію систем нечіткого логічного висновку з пакетом Simulink:

1. **fuzblock** – модулі контролерів на основі нечіткого логічного висновку;
2. **sffis** – функція виконання нечіткого логічного висновку, оптимізована під Simulink.

Четверта категорія програмних інструментів пакета Fuzzy Logic Toolbox включає такі демонстраційні приклади:

1. **defuzzdm** – дефаззифікація різними методами;
2. **fcmdemo** – 2D-кластеризація з використанням алгоритму fuzzy c-means;
3. **fuzdemos** – список всіх демонстраційних прикладів Fuzzy Logic Toolbox;
4. **gasdemo** – використання алгоритму ANFIS і субтрактивної кластеризації для ідентифікації залежності паливної ефективності (витрати палива на одну милю) від шести параметрів автомобіля;
5. **invkine** – інверсна кінематика робота-маніпулятора;
6. **irisfcm** – використання алгоритму fuzzy c-means для кластеризації ірисів;
7. **juggler** – жонглювання шариком з допомогою тенісної ракетки з демонстрацією нечіткої бази знань;
8. **noisedm** – адаптивне подавлення шумів;
9. **slbb** – управління системою «шарик на коромислі» (необхідний пакет Simulink);
10. **slcp** – управління системою «перевернутий маятник» (необхідний пакет Simulink);
11. **sltank** – управління рівнем води (необхідний пакет Simulink);
12. **sltankrule** – управління рівнем води з демонстрацією нечіткої бази знань (необхідний пакет Simulink);
13. **sltbu** – парковка вантажного авто (необхідний пакет Simulink);
14. **mgtsdemo** – передмовлення часового ряду Маккея-Глесса;
15. **trips** – побудова моделі прогнозування кількості автомобільних поїздок;
16. **shower** – управління душем;
17. **slcp1** – переміщення нестійкої системи «перевернутий маятник змінної довжини на возику» в задану точку;
18. **slcpp1** – переміщення нестійкої системи «два перевернутих маятника змінної довжини на возику» в задану точку;
19. **mfdemo** – виведення на екран вікна, що включає графіки всіх запрограмованих в Fuzzy Logic Toolbox типів функцій належності;
20. **drydemo** – ілюстрація використання технології ANFIS для ідентифікації нелінійних динамічних систем на прикладі процесу нагрівання повітря в фені.

5.3.2 Графічний інтерфейс редактора нечіткої системи висновку

Fuzzy Logic Toolbox за допомогою програм, що входять до його складу, реалізує графічний інтерфейс на основі:

- Редактора нечіткої системи висновку Fuzzy Inference System Editor (FIS Editor або FIS–редактор);
- Редактора функцій належності (Membership Function Editor);
- Редактора правил (Rule Editor);
- Переглядача правил (Rule Viewer);
- Переглядача поверхні відгуку (Surface Viewer).

Розглянемо послідовність дій за створенням нечіткої системи, тобто нечіткого логічного контролера в програмному пакеті **Fuzzy Logic Toolbox**.

Командою Fuzzy в режимі командної строки запускається основна інтерфейсна програма Fuzzy Logic – редактор нечіткої системи висновку, вид вікна інтерфейсу якої показано на рис.5.9.

Стрічка меню включає такі позиції: **File** – робота з файлами моделі (створення, збереження, виведення та друк); **Edit** – операції редагування (додавання та виключення вхідних та вихідних змінних); **View**– переходи до додаткових редакторів.

Для створення нечіткої системи виберемо команди **New FIS > Mamdani** з меню **File**. У білому середньому квадраті блок–схеми моделі з'явиться напис **Untitled2**. Змінимо назву моделі на **b1**, виконавши експорт моделі на диск з іменем **b1**.

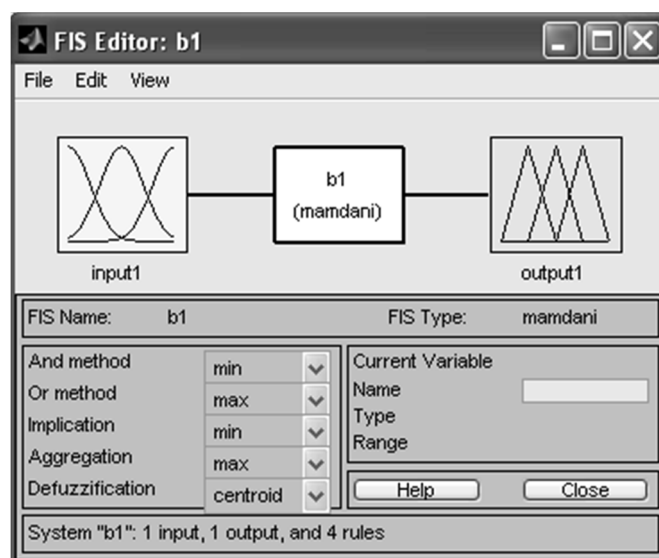


Рис. 5.9. Основне вікно редактора Fuzzy Logic

Натискаємо один раз лівою кнопкою маніпулятора на лівому блоці. Назва цього блока відображається в полі вікна **Name** в правій нижній частині інтерфейсного вікна редактора. За необхідності назву вхідної змінної можна

змінити. Також можна додати за допомогою команди **Add Variable > Input** з меню **Edit** ще одну (чи більше) вхідну змінну. Кількість лінгвістичних змінних теоретично не обмежена, але для зменшення складності побудови системи відношень нечітких змінних бази знань та підвищення швидкодії системи достатньо використовувати не більше п'яти вхідних та двох вихідних змінних.

Таким же чином, як для вхідної змінної, за допомогою команди **Add Variable > Output** з меню **Edit** додається і вихідна змінна.

Подвійне натискання кнопки на блоці вхідної змінної призводить до відкриття редактора функцій належності – Membership Function Editor (рис.5.10).

Для створення терм-множини вхідної змінної Input1 відкриваємо меню **Edit** цього редактора та командою **Add MFs** вводимо або додаємо функції належності (ФН) та їх параметри (діапазон вхідної змінної, назви ФН (терми), тип ФН та координати характерних точок). Для ФН трикутної або трапецієподібної форми характерними точками будуть координати їх вершин за віссю вхідної змінної. Після введення всіх параметрів необхідно натиснути ліву кнопку маніпулятора в довільній точці вікна редактора або підтвердити введення параметрів натисканням кнопки **OK** на клавіатурі.

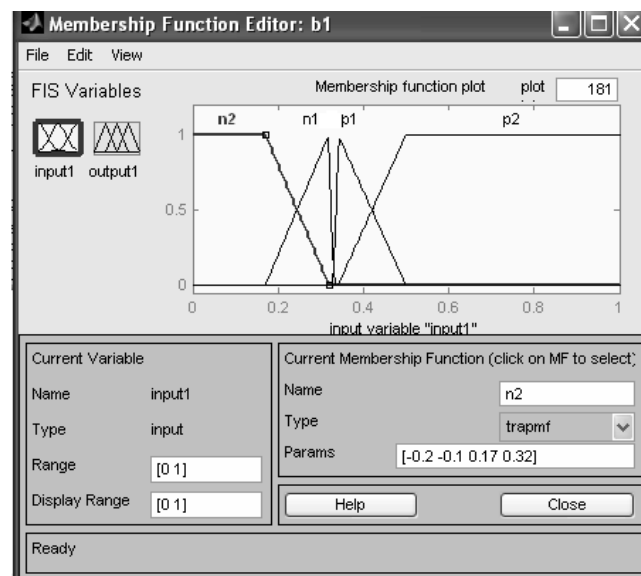


Рис.5.10. Вікно редактора функцій належності – Membership Function Editor вхідної лінгвістичної змінної з термами: n2, n1, p1, p2

Необхідно відмітити, що перед введенням координат ФН у полі **Range**, необхідно установити діапазон вхідної змінної. Як правило, діапазон задається у відносних одиницях від -1 до 1 або від 0 до 1 , але не виключається введення діапазону в абсолютних значеннях. Введення відносних одиниць дає змогу легко масштабувати змінну та транспонувати контролер в інші системи управління подібного типу.

Коригування координат характерних точок ФН можна виконувати їх перетягуванням в графічному режимі шляхом активації функції (зміна кольору з чорного на червоний) лівою кнопкою маніпулятора з фіксацією натискання та поступовим пересуванням точок. Координати точок автоматично будуть змінюватися в полі **Params**. Більш точне числове налаштування можна виконати в полі **Params**. На вкладці **Current Membership Function** в полях **Name** та **Type** задаються або коригуються ім'я та тип ФН.

Таким же чином, як і для вхідної змінної, подвійним натисканням лівої кнопки на блоці вихідної змінної запускаємо редактор функцій належності – Membership Function Editor (рис. 5.11) та командою **Add MFs** з меню **Edit** цього редактора вводимо або добавляємо функції належності (ФН) і їх параметри (діапазон вихідної змінної, назви ФН (терми), тип ФН та координати характерних точок).

Наступним кроком синтезу нечіткого контролера після того як введені параметри ФН є введення правил. Для цього повертаємось до вікна FIS–редактора (рис.5.9) та два рази клацаємо лівою кнопкою маніпулятора на середньому (білому) блоці і запускаємо редактор правил – **Rule Editor** (рис. 5.12). Поле правил, що знаходиться вверху вікна редактора правил спочатку після запуску, є пустим. Введення правила виконується маркуванням відповідного вхідного терма, що входить до умови правила, шляхом одноразового натискання лівою кнопкою маніпулятора на його імені в полі переліку вхідних термів (поле зліва) та відповідним маркуванням вихідного терма цього правила в правому полі з переліком вихідних термів. Далі в полі **Weight**, що знаходиться в нижній частині вікна, вводиться ваговий коефіцієнт правила в інтервалі від 0 до 1. Якщо контролер має декілька вхідних чи вихідних змінних, то у вікні редактора.

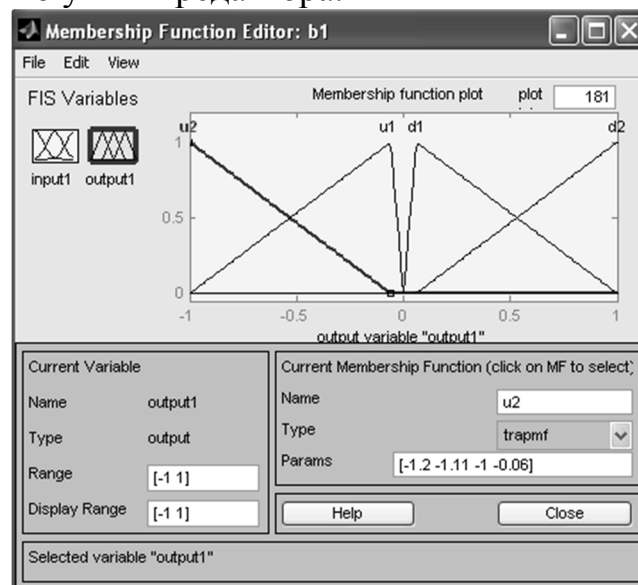


Рис. 5.11. Вікно редактора функцій належності – Membership Function Editor вихідної лінгвістичної змінної з термами: u2, u1, d1, d2

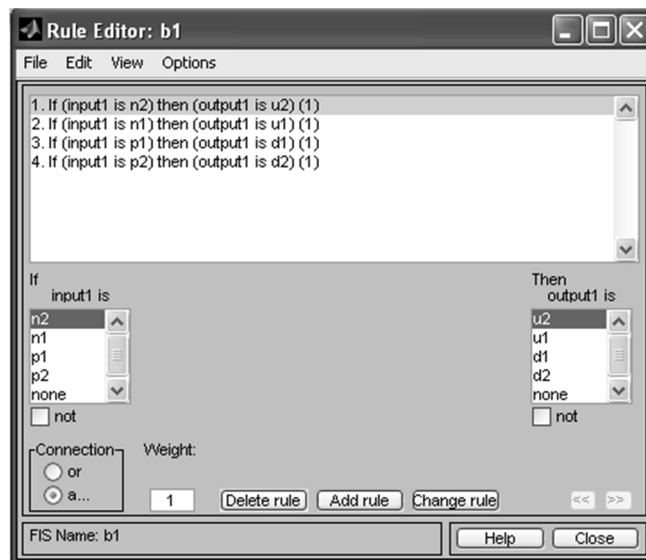


Рис. 5.12. Вікно редактора правил – **Rule Editor**

додатково з'являються поля з переліком термів цих змінних, які використовуються для побудови правила. Окрім цього у разі декількох вхідних змінних, тобто за розгалуженої умови правила необхідно на вкладці **Connection** активувати радіокнопки «**Or**» чи «**And**», що символізують відповідно операції об'єднання «Або» та «І» підумов. Далі, після виконаних дій необхідно натиснути лівою кнопкою маніпулятора на кнопці **Add rule**. Додане правило з'явиться у верхньому полі вікна редактора. За необхідності введені правила можна видалити або модифікувати, використовуючи кнопки **Delete rule** та **Change rule**.

Після введення параметрів ФН та формування правил бази знань необхідно визначити опції процедури (алгоритму) формування нечіткого логічного висновку шляхом введення їх операторів, які і будуть формувати значення вихідного параметра. Опції надані вкладками з полями в нижній лівій частині вікна редактора. До них відносяться:

1. **AND та OR method (Aggregation)** – методи, що визначають процес об'єднання підумов, які входять до умови (лівої частини) правил (якщо вони мають місце) і формування ступеня правдивості (виконання) умови;
2. **Implication** метод, що визначає процес формування ступеня виконання вихідної частини правила (висновку правила) за ступенем правдивості умови, тобто процес активації правил. Ті правила, ступінь правдивості яких відрізняється від нуля, вважаються активними і використовуються для подальших розрахунків;
3. **Accumulation** метод, що визначає процес об'єднання нечітких висновків активних правил кожної лінгвістичної змінної;

4. **Defuzzification** метод, що визначає процес переходу від нечіткого логічного рішення до значення фізичного вихідного параметра моделі контролера.

Кожна з перерахованих опцій редактора має своє вікно, що з'являється у разі натискання лівою кнопкою мишки на трикутнику в правій частині відповідного поля, призначена для введення таких операторів:

1. **AND method:** **Min** – метод мінімального значення,
Prod – метод алгебраїчного добутку,
Custom – метод, визначений користувачем;
2. **OR method:** **Max** – метод мінімального значення,
Probor – метод алгебраїчної суми,
Custom – метод, визначений користувачем;
3. **Implication:** **Min** – метод мінімального значення,
Prod – метод алгебраїчного добутку,
Custom – метод, визначений користувачем;
4. **Accumulation:** **Max** – метод мінімального значення,
Sum – метод обмеженої суми,
Probor – метод алгебраїчної суми,
Custom – метод, визначений користувачем;
5. **Defuzzification:** **Centroid** – метод центра ваги;
Bisector – метод центра площі;
Mom – метод середнього максимуму;
Lom – метод найбільшого (правого) модального значення;
Som – метод найменшого (лівого) модального значення;
Custom – метод, визначений користувачем.

Взагалі у разі початкового моделювання системи оператори процедури нечіткого рішення в програмному пакеті можна залишити установленими за замовчуванням. Вони, як правило, є досить прийнятними.

Після закінчення введення операторів можна вважати, що побудова нечіткого контролера закінчена і можна зберегти на диску синтезовану модель системи командою **Export > to Disk** або передати в робочу область імітаційної моделі командою – **to Workspace** з меню **File** вікна редактора для виконання моделювання системи.

Наглядну візуальну оцінку механізму формування нечіткого висновку через перегляд графічної інформації активації правил можна виконати командами **View > Rules**. На рис. 5.13 графічно представлено процес активації чотирьох правил моделі b1 нечіткого логічного контролера у разі числового значення вхідної базової змінної на рівні 0,5.

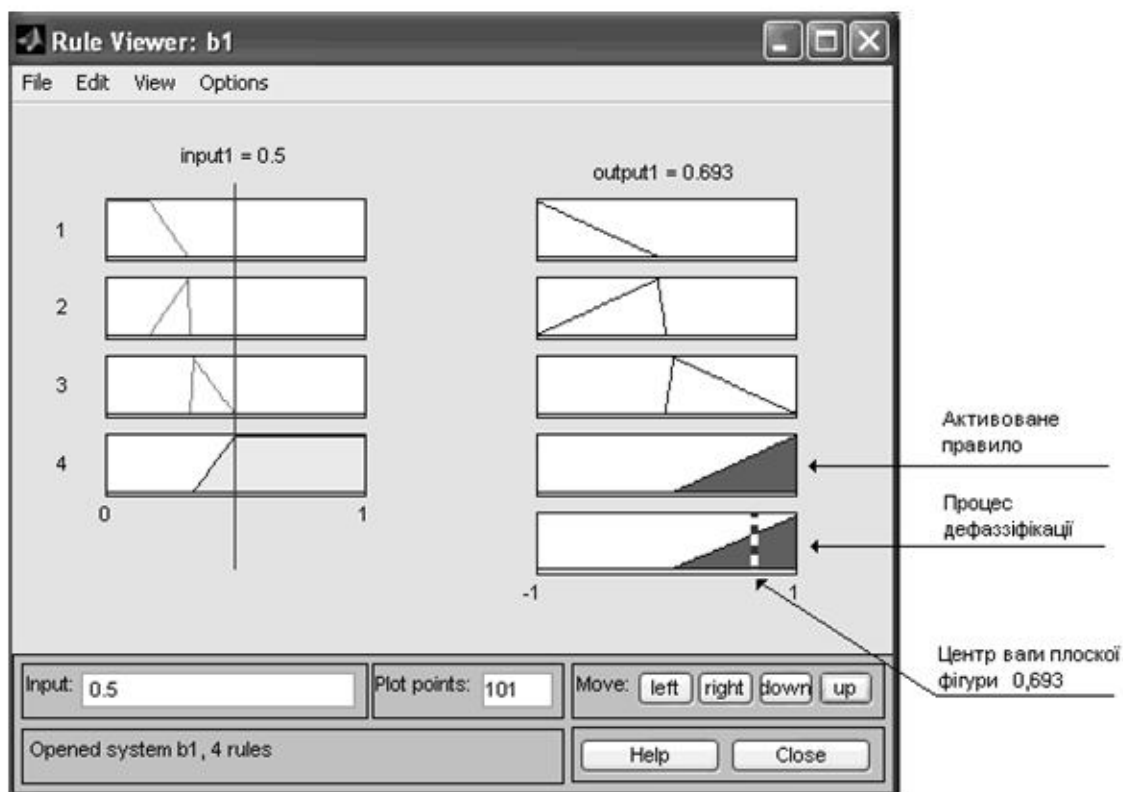


Рис. 5.13. Процес активації чотирьох правил моделі b1

Як видно з рис. 5.13, у процесі нечіткого логічного висновку активовано тільки одне правило (нижнє), яке дало нам у процесі процедури накопичення площу плоскої фігури (темний прямокутник). У процесі дефаззифікації цієї фігури методом центру ваги, що заданий як **Centroid** на рис. 4.37, визначається числове значення 0,693 вихідного параметра моделі. Переміщуючи за допомогою стрілки ручного маніпулятора вертикальну пряму чи задаючи в полі **input** числове значення вхідного параметра (рис. 5.12), в полі вихідних функцій належності можна спостерігати активацію правил та формування значення вихідного параметра.

На рис.5.14 наведена графічна інформація формування вихідного параметра за значення **input1=0,393**. З рисунка видно, що в цьому випадку активовано два правила: третє та четверте. Площі фігур трапецій, що символізують рішення за цими правилами, отримані в процесі логічного висновку (**Implication**), після процесу накопичення або акумуляції (**Accumulation**) методом **Max** (рис. 4.31) дають плоску фігуру, що зображена внизу справа на рис.5.14. У результаті формування рішення управління вихідний параметр **output1** дорівнює 0,436.

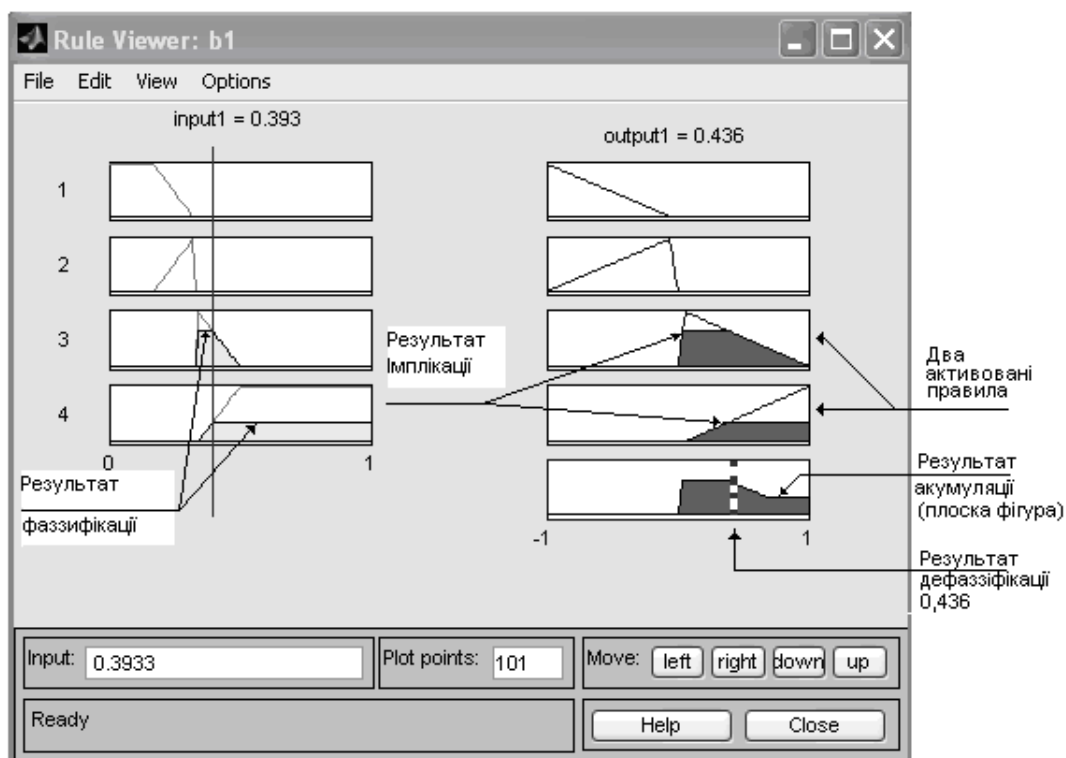


Рис. 5.14. Графічна інформація формування вихідного параметра за значення **input1=0,393**

У полі **Plot points** задається кількість точок формування для кожної з кривих функцій належності. Кнопки поля **Move** використовуються для зміщення кривих за полем вікна редактора.

Таким чином, аналіз результатів нечіткого висновку за різних значень вхідних змінних дає змогу оцінити вихідний сигнал управління, тобто реакцію розробленої моделі, та визначити адекватності її дій у реальній системі. На гнучкість та адекватність реакції системи суттєво впливає як кількість вхідних та вихідних параметрів, так і адекватне відображення їх нечіткими змінними.

Виходячи з візуальної оцінки моделі, проводиться модифікація заданих правил, їх вагових коефіцієнтів чи введення нових правил. За необхідності проводиться корекція функцій належності (форми, діапазону нечіткої множини, потужності множини ФН) вхідних та вихідних змінних.

Як один з шляхів за формування користувачем самостійно функцій належності нечітких змінних, тобто установлення ступеня належності певного значення базової фізичної змінної (параметра) тій чи іншій нечіткій множині, на початковому етапі синтезу нечіткої моделі можна рекомендувати використання штучно введених рангів, що виражають значимість фізичного значення певного параметра в порівнянні з опорним його значенням за формування властивості, що виражає нечітка змінна на терм-множині лінгвістичної змінної. Опорними значеннями у разі установлення рангів можуть

бути характерні (номінальні, мінімальні, максимальні) значення певних діапазонів цих параметрів. Функції належності, визначені користувачем, мають бути представлені у вигляді **m**-файлів і знаходитись в папці ***:\MATLAB...\toolbox\fuzzy**.

Важливим моментом дослідження нечіткої моделі є її передатна характеристика (коефіцієнт відгуку) або для систем з декількома вхідними параметрами – поверхня управління (відгуку). Для побудови передатної характеристики використовується спеціальний редактор, який запускається командою **View > Surface**. Графічний інтерфейс програми представлено на рис. 5.15. За отриманим графіком передатної функції можна оцінити вплив зміни значень вхідних параметрів на значення вихідного чи вихідних параметрів нечіткої моделі. Для цього в нижній частині вікна необхідно вибрати змінну в списку поля, що розкривається **X(input)**, а в списку поля **Y(output)** вибрати **none**. Отриману передатну функцію моделі **b1** представлено на рис. 5.15. Необхідно відмітити, що на характер форми кривої передатної характеристики впливають, насамперед, інтенсивність розміщення функцій належності та відповідне їх перемещування, а на саме числове значення коефіцієнта відгуку системи більший вплив має форма ФН.

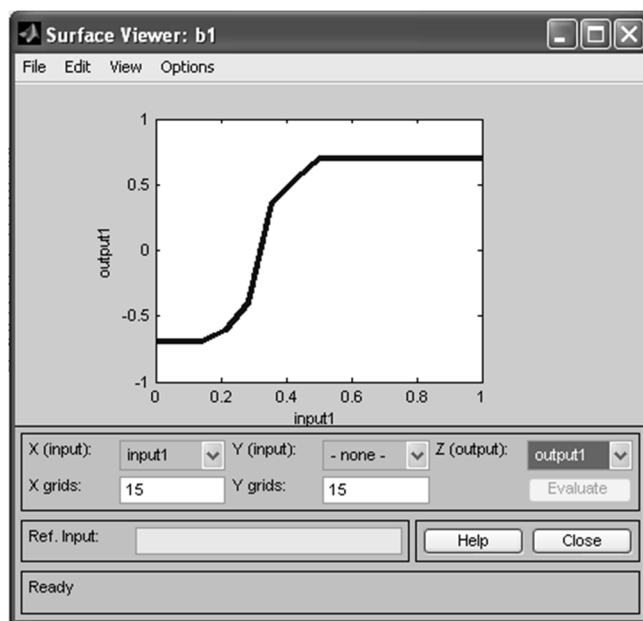


Рис. 5.15. Графічний інтерфейс програми формування передатної характеристики нечіткої моделі

Для збереження створеної нечіткої моделі необхідно скористатися командою **File > Export > to disk**, у цьому разі на диску створюється (ASCII) файл звичайного текстового формату з розширенням ***.fis**. Його можна переглядати та редагувати текстовим редактором без системи MATLAB, а також використовувати повторно в наступних сеансах роботи з системою.

Збереження моделі командою **File > Export > to workspace** дає змогу «легалізувати» створену модель під заданим іменем у робочому середовищі протягом поточного сеансу роботи та не допускає її повторного використання в наступному сеансі роботи MATLAB.

Досить часто виникає необхідність використання функцій збереження, відкриття та розрахунку створеної системи нечіткого висновку в режимі командної строки, як наприклад:

```
readfis('ім'я файла');  
evalfis(вектор параметрів, ім'я);  
writefis(ім'я) або writefis(ім'я, 'ім'я файла').
```

У наведених командах *'ім'я файла'* – це найменування файлу з записаною системою (без розширення), *ім'я* – ідентифікатор, що присвоєний системі в робочому середовищі MATLAB, *вектор параметрів* – набір значень входів, для яких необхідно розрахувати вихід.

Часто за оформлення та редагування результатів необхідно використовувати графічне вікно. Розглянемо приклад, коли модель під іменем **b1** записана на диску **d:\Matlab\toolbox\fuzzy\b1**.

Виконаємо зчитування моделі в робоче середовище під іменем *a*

```
a=readfis('d:\Matlab\toolbox\fuzzy\b1') ↓  
a =  
      name: 'b1'  
      type: 'mamdani'  
      andMethod: 'min'  
      orMethod: 'max'  
      defuzzMethod: 'centroid'  
      impMethod: 'min'  
      aggMethod: 'max'  
      input: [1x1 struct]  
      output: [1x1 struct]  
      rule: [1x4 struct].
```

Графічне вікно MATLAB з мнемонічним зображенням розробленої моделі відкривається командою

```
plotfis(a).
```

Графічне вікно MATLAB з графіками функцій належності відкривається командою (рис. 5.16)

```
plotmf(a,'input',1).
```

Графічне вікно MATLAB з поверхнею відгуку відкривається командою

```
gensurf(a).
```

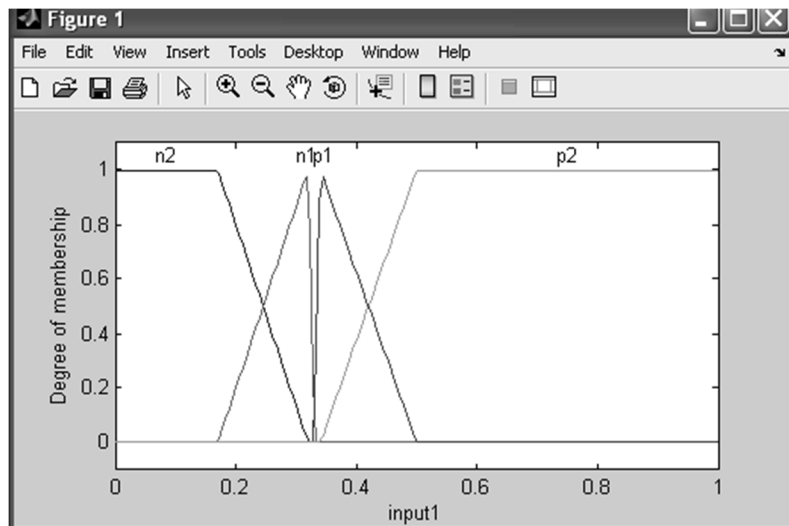


Рис.5.16. Результат виконання команди `plotmf(a,'input',1)`

Інтерфейс графічного вікна рис. 5.15 має достатньо команд для редагування графіків та трансляції їх в графічні та текстові редактори (наприклад, Microsoft Word).

КОНТРОЛЬНІ ЗАПИТАННЯ ТА ЗАВДАННЯ

1. Основні та найбільш важливі критерії, яким пристрій керування з нечітким алгоритмом має задовольняти.
2. Принципи формування функцій належності лінгвістичних змінних.
3. Основні джерела та шляхи формування правил нечіткого логічного контролера.
4. Методи рішення можливих проблем систем управління з нечітким логічним контролером.
5. Поверхня управління та її формування.
6. Операції виконання алгоритму нечіткого логічного висновку.
7. Вибір кількості функцій належності лінгвістичних змінних.
8. Формування правил нечіткого логічного контролера.
9. Методи дефазифікації нечіткого логічного висновку.

ЛІТЕРАТУРА

[22], стор. 302–313; [34], стор. 26–36; [12], стор. 208–228

РОЗДІЛ 6

НЕЧІТКИЙ КЛАСТЕРНИЙ АНАЛІЗ ДАНИХ

6.1 Загальна постановка задачі кластерного аналізу

Терміном кластерний аналіз прийнято позначати сукупність методів, підходів і процедур, розроблених для вирішення проблеми формування однорідних класів у довільній проблемній області. Необхідність аналізу великих обсягів об'єктивної і суб'єктивної інформації, пов'язаних з неформалізованими і погано формалізованими завданнями різного фізичного походження, зажадала інтенсивного розвитку нових наукових напрямків, серед яких важливу роль відіграють прикладна статистика і методи аналізу даних.

Методи аналізу даних, складовою частиною яких є методи кластерного аналізу, навпаки, не використовують апріорних припущень про ймовірнісну природу початкової інформації і керуються тільки евристичними міркуваннями про характер і особливості досліджуваної сукупності об'єктів.

Кластерний аналіз (автоматична класифікація, розпізнавання режимів, розпізнавання образів, чисельна таксономія) займає одне із центральних місць серед методів аналізу даних і представляє сукупність підходів, методів і алгоритмів, призначених для знаходження деякого розбиття досліджуваної сукупності об'єктів на підмножини щодо подібних, схожих між собою об'єктів. Об'єкти з одного кластеру повинні мати більшу схожість між собою, ніж з об'єктами інших кластерів.

Виявлення або знаходження кластерів у множині даних досліджуваної сукупності має відповідати таким вимогам:

- кожен кластер має представляє собою концептуально однорідну категорію і містити схожі об'єкти з близькими значеннями властивостей або ознак;
- сукупність всіх кластерів має бути вичерпною, тобто охоплювати всі об'єкти досліджуваної сукупності.

Формально під завданням кластерного аналізу заданої множини об'єктів розуміється завдання знаходження деякого теоретико–множинного розбиття (групування, покриття) цієї вихідної множини об'єктів на підмножини таким чином, щоб елементи, що відносяться до однієї підмножини, відрізнялися між собою значно меншою мірою, ніж елементи з різних підмножин. Підмножини, що володіють подібною властивістю, також називаються кластерами.

Методи і алгоритми кластерного аналізу, як інструмент попереднього або розвідувального аналізу даних, незамінні при пошуку закономірностей у великих обсягах багатовимірних даних, таких як сховища даних. При цьому

проблема кластер-аналізу набуває самостійного значення в контексті інтелектуального аналізу даних (Data Mining).

Концептуальний взаємозв'язок між кластерним аналізом і теорією нечітких множин базується на тій обставині, що у разі вирішення задач структуризації складних систем більшість класів об'єктів, які формуються, розмиті за своєю природою. Ця розмитість полягає в тому, що перехід від належності до неналежності елементів до даних класів швидше поступовий, ніж стрибкоподібний.

Так, у загальному випадку завданням нечіткої кластеризації є знаходження нечіткого розбиття або нечіткого покриття множини елементів досліджуваної сукупності, які утворюють структуру нечітких кластерів, присутніх у розглянутих даних. Це завдання зводиться до знаходження ступенів належності елементів універсуму шуканим нечітким кластерам, які в сукупності і визначають нечітке розбиття або нечітке покриття початкової множини розглянутих елементів.

Нехай досліджувана сукупність даних є кінцевою множиною елементів $A = \{a_1, a_2, \dots, a_n\}$, яка отримала назву множини об'єктів кластеризації. До розгляду також вводиться кінцева множина ознак або атрибутів $P = \{p_1, p_2, \dots, p_q\}$, кожна з яких кількісно представляє деяку властивість або характеристику елементів розглянутої проблемної області. У цьому разі натуральне n визначає загальну кількість об'єктів даних, а натуральне q – загальну кількість вимірних ознак об'єктів.

Далі передбачається, що для кожного з об'єктів кластеризації деяким чином виміряні всі ознаки множини P в деякій кількісній шкалі. Тим самим кожному з елементів $a_i \in A$ поставлений у відповідність деякий вектор $X_i = (x_1^i, x_2^i, \dots, x_q^i)$, де x_j^i – кількісне значення ознаки $p_j \in P$ для об'єкта даних $a_i \in A$. Для визначеності будемо вважати, що всі x_j^i набувають деяких дійсних значень, тобто $x_j^i \in R$.

Процес представлення властивостей може бути реалізований у різних шкалах, кожна з яких характеризується допустимим перетворенням даних.

Розрізняють кілька типів шкал:

1. *Шкала найменувань* або номінальна шкала. Може бути використана для встановлення відношення еквівалентності елементів щодо розглянутої ознаки. У цьому випадку в процесі вимірювання деякої ознаки об'єкта ставиться у відповідність деякий символ або номер, який лише відрізняє одне значення ознаки від другого. Допустимим перетворенням в шкалах найменувань, яке не спотворює результати вимірювання ознак в цій шкалі, є бієктивне або взаємно однозначне відображення між двома множинами значень ознак. Бінарну шкалу, яка складається з двох елементів, позначених довільними символами, наприклад: $\{0, 1\}$ або $\{+, -\}$, зручно вважати окремим випадком шкали найменувань. *Приклади*

ознак, вимірюваних в шкалах найменувань: стать людини, марка автомобіля, назви вулиць, міст і інших об'єктів.

2. *Порядкова шкала*, або шкала порядку. Дає змогу встановити відношення порядку елементів щодо розглянутої ознаки. У цьому випадку в процесі вимірювання ознаки об'єкта ставиться у відповідність, як правило, деяке натуральне або ціле число, яке може бути інтерпретоване як значення ознаки в балах. Допустимим перетворенням в шкалах порядку, яке не спотворює результати вимірювання ознак в цій шкалі, є довільне монотонно зростаюче відображення чи функція між двома множинами значень ознак. *Приклади ознак, вимірюваних в порядкових шкалах: бали або оцінки на іспитах, бали за виступи в деяких видах спорту.*
3. *Інтервальна шкала*, або шкала інтервалів. Дає змогу встановити порівняння інтервалів значень розглянутої ознаки. У цьому випадку в процесі вимірювання ознаки об'єкта ставиться у відповідність, як правило, деяке дійсне число, рівне значенню цієї ознаки. Допустимим перетворенням в шкалах інтервалів є довільна лінійна зростаюча функція між двома множинами значень ознак. Характерною властивістю цієї шкали є відсутність абсолютного нуля. *Приклад ознаки, що вимірюється в інтервальній шкалі: температура в шкалах Цельсія і Фаренгейта.*
4. *Шкала відношень*. Дає змогу встановити порівняння відношень значень розглянутої ознаки. У цьому випадку в процесі вимірювання ознаки об'єкта ставиться у відповідність також деяке дійсне число, що дорівнює значенню цієї ознаки. Допустимим перетворенням в шкалах відношень є довільна лінійна зростаюча функція, що проходить через нуль. Характерною властивістю цієї шкали є наявність абсолютного нуля. *Приклади ознак, вимірюваних в шкалі відношень: відстань у метрах, маса в кілограмах, швидкість в км/год.*

6.2 Нечітка кластеризація і алгоритм її реалізації

Множину ознак кластерів слід вибирати так, щоб всі x_j^i були виміряні в шкалі відношень або шкалі інтервалів. Саме в цьому випадку результати нечіткої кластеризації мають змістовну інтерпретацію, адекватну проблемі знаходження нечітких кластерів.

Вектори значень ознак $X_i = (x_1^i, x_2^i, \dots, x_q^i)$ зручно представити у вигляді так званої матриці даних D розмірності $(n \times q)$, кожен рядок якої дорівнює значенням вектора X_i .

Загальне завдання нечіткого кластерного аналізу формулюється в такий спосіб: на основі початкових даних D визначити таке нечітке розбиття $\mathcal{R}(\tilde{A}) = \{\tilde{A}_k | \tilde{A}_k \subseteq \tilde{A}\}$ або нечітке покриття $M(\tilde{A}) = \{\tilde{A}_k | \tilde{A}_k \subseteq \tilde{A}\}$ множини $\tilde{A} = A$ на задане число c нечітких кластерів $\tilde{A}_k (k \in \{2, \dots, c\})$, яке доставляє

екстремум деякої цільової функції $f(\mathcal{R}(\tilde{A}))$ серед всіх нечітких розбиттів або екстремум цільової функції $f(M(\tilde{A}))$ серед всіх нечітких покриттів.

Для конкретизації завдання нечіткого кластерного аналізу може бути використана спеціальна функція fcm системи MATLAB, основана на методі нечітких c -Means. Для уточнення виду цільової функції $f(M(\tilde{A}))$ до розгляду вводяться деякі додаткові поняття. Перш за все передбачається, що шукані нечіткі кластери є нечіткими множинами $\tilde{A}_k$, що утворюють нечітке покриття початкової множини об'єктів кластеризації $\tilde{A}=A$, для якого має місце умова такого вигляду:

$$\sum_{k=1}^c \mu_{A_k}(a_i) = 1 \quad (\forall a_i \in A), \quad (6.1)$$

де c – загальна кількість нечітких кластерів $\tilde{A}_k (k \in \{2, \dots, c\})$, яка вважається попередньо заданою ($c \in N$ та $c > 1$)

Зауваження

Необхідність умови (6.1) обумовлюється тією обставиною, що шукане нечітке покриття має «покривати» звичайну не нечітку множину об'єктів кластеризації A , яка в той же час є нечіткою множиною $\tilde{A}$, для якої значення функцій належності кожного з елементів дорівнює одиниці. Іноді нечітке покриття, що задовольняє умові (6.1), називають нечітким розбиттям в сенсі Заде.

Для кожного нечіткого кластера вводяться до розгляду так звані *типові представники* або *центри* v_k шуканих нечітких кластерів $\tilde{A}_k (k \in \{2, \dots, c\})$, які розраховуються для кожного із нечітких кластерів і за кожною з ознак (параметрів) за такою формулою:

$$v_j^k = \frac{\sum_{i=1}^n (\mu_{A_k}(a_i))^m \cdot x_j^i}{\sum_{i=1}^n (\mu_{A_k}(a_i))^m} \quad (\forall k \in \{2, \dots, c\}, \forall p_j \in P), \quad (6.2)$$

де m – деякий параметр, що називається експоненціальною вагою і дорівнює деякому дійсному числу ($m > 1$). Кожен із центрів кластерів є вектором $v_k = (v_1^k, v_2^k, \dots, v_q^k)$ в деякому q -мірному нормованому просторі, ізоморфному R^q , тобто $v_j^k \in R^q$, якщо всі ознаки виміряні в шкалі відношень.

В якості цільової функції розглядається сума квадратів зважених відхилень координат об'єктів кластеризації від центрів шуканих нечітких кластерів:

$$f(A_k, v_j^k) = \sum_{i=1}^n \sum_{k=1}^c (\mu_{A_k}(a_i))^m \sum_{j=1}^q (x_j^i - v_j^k)^2, \quad (6.3)$$

де m – експоненціальна вага нечіткої кластеризації ($m \in R, m > 1$), значення якої задається залежно від кількості елементів (потужності) множини A . Чим більше елементів містить множина A , тим менше значення вибирається для m .

Алгоритм нечіткої кластеризації може бути сформульований таким чином: для заданих матриці даних D , кількості нечітких кластерів c ($c \in N$ та $c > 1$), параметра m визначити матрицю U значень функцій належності об'єктів кластеризації $a_j \in A$ нечітким кластерам $\tilde{A}_k$ ($k \in \{2, \dots, c\}$), які задовольняють мінімуму цільовій функції (6.3) і задовольняють обмеженням (6.1), (6.2), а також додатковим обмеженням (6.4) і (6.5):

$$\sum_{i=1}^n \mu_{A_k}(a_i) > 0 \quad (\forall k \in \{2, \dots, c\}); \quad (6.4)$$

$$\mu_{A_k}(a_i) \geq 0 \quad (\forall k \in \{2, \dots, c\}, \forall a_i \in A). \quad (6.5)$$

Умова (6.4) виключає появу порожніх нечітких кластерів у шуканій нечіткій кластеризації. Умова (6.5) має суто формальний характер, оскільки безпосередньо впливає з визначення функції належності нечітких множин. У цьому випадку цільова функція (6.3) мінімізує відхилення всіх об'єктів кластеризації від центрів нечітких кластерів пропорційно значенням функцій належності цих об'єктів відповідним нечітким кластерам.

Оскільки цільова функція (6.2) не є опуклою, а обмеження (6.1), (6.4), (6.5) в своїй сукупності формують неопуклу множину допустимих альтернатив, то в загальному випадку задача нечіткої кластеризації відноситься до багатоекстремальних завдань нелінійного програмування.

Перевагою постановки задачі нечіткої кластеризації у вигляді (6.1) – (6.5) є природна інтерпретація як шуканих нечітких кластерів, що визначаються функціями належності (6.5), так і їх типових представників або центрів (6.2), які також визначаються в результаті рішення поставленого завдання.

Недоліком такої постановки завдання нечіткої кластеризації є необхідність апіорного встановлення загального числа нечітких кластерів c ($c \in N$ та $c > 1$), яке в окремих випадках може бути невідомим. Ця обставина може зажадати залучення додаткових процедур для його визначення або вирішення поставленого завдання для кількох значень c з подальшим вибором найбільш адекватного результату нечіткої кластеризації.

6.3 Рішення завдання нечіткої кластеризації з використанням засобів графічного інтерфейсу в програмному додатку Fuzzy Logic Toolbox

У системі MATLAB для рішення задачі нечіткої кластеризації на основі алгоритму FCM (Fuzzy *c*–Means) може бути використана спеціальна функція командного рядка *fcm* або спеціальний графічний інтерфейс кластеризації, що викликається функцією *findcluster*.

Для прикладу розглянемо множину даних, які містяться в системі MATLAB і використовуються в якості тестової сукупності об'єктів нечіткої кластеризації. Ці дані представляють собою матрицю даних *D* розмірності (140x2), що знаходяться в файлі *fcmdata.dat*, який поставляється разом з системою MATLAB. У цьому випадку матриця даних відповідає 140 об'єктам, для кожного з яких виконані вимірювання за двома ознаками, що є зручним для візуалізації початкових даних і результатів нечіткої кластеризації в двовимірному просторі на площині. Графік координат точок на площині, відповідних об'єктам нечіткої кластеризації, зображений на рис. 6.1.

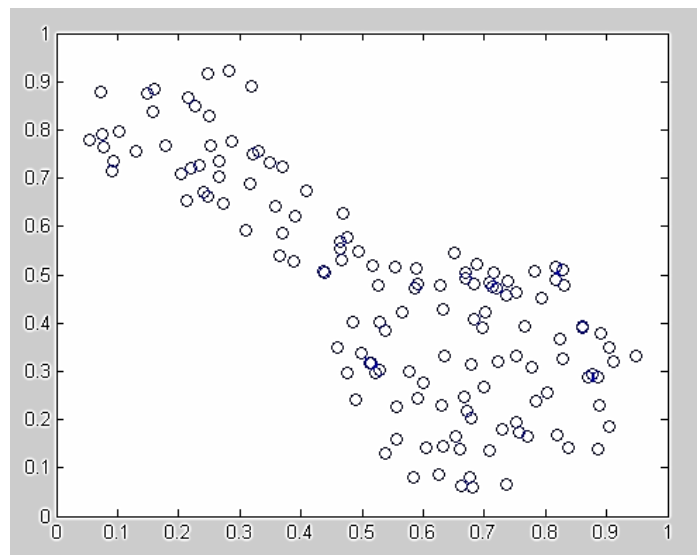


Рис. 6.1. Множина об'єктів нечіткої кластеризації, відповідних матриці даних з файлу *fcmdata.dat*

Результат розв'язання задачі нечіткої кластеризації для трьох нечітких кластерів з використанням зазначеної послідовності команд може бути візуалізований (рис. 6.2).

Функція *findcluster* призначена для виклику графічного інтерфейсу програми нечіткої кластеризації для методу нечітких *c*–Means і методу субтрактивної кластеризації (*subtractive clustering*). Вона може бути викликана одним із наступних форматів: *findcluster* або *findcluster ('file. dat')*.

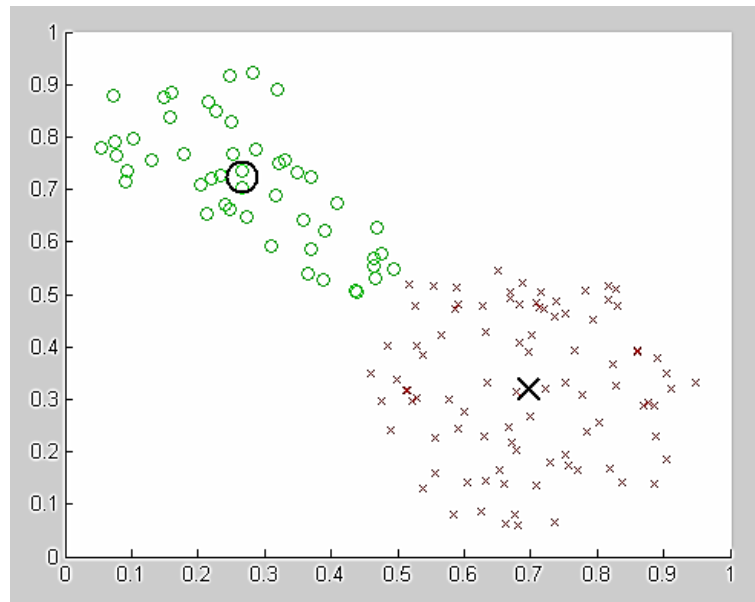


Рис. 6.2. Результат розв'язання задачі нечіткої кластеризації для матриці даних з файлу *fcmdata.dat*

У першому випадку функція *findcluster* викликає графічний інтерфейс GUI програми для виконання нечіткої кластеризації алгоритмом FCM або нечіткої субтрактивної кластеризації. У цьому разі необхідно завантажити в робочу область початкові дані з зовнішнього файлу за допомогою кнопки *Load Data* (рис. 6.3).

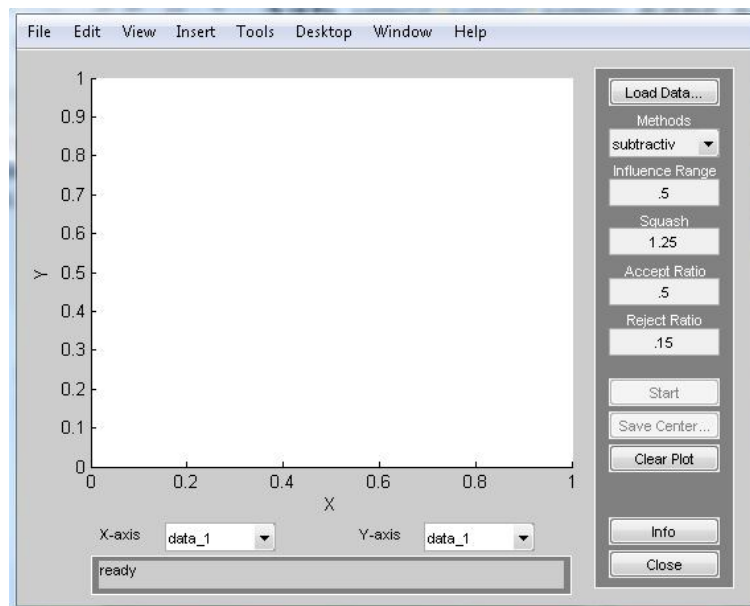


Рис. 6.3. Вибір зовнішнього файлу для завантаження початкових даних в робочу область

Вибір методу нечіткої кластеризації здійснюється за допомогою списку *Methods*, що розкривається. Для кожного з методів нечіткої кластеризації у відповідних рядках введення встановлені значення параметрів алгоритмів за замовчуванням. Ці значення можуть бути змінені користувачем. Для цього необхідно встановити курсор введення у відповідне поле і набрати потрібні цифри з урахуванням допустимих значень параметрів.

У другому випадку функція *findcluster ('file. Dat')* викликає графічний інтерфейс, а в робочу область автоматично завантажуються дані кластеризації із зовнішнього файлу *file.dat*. У цьому разі на графіку відображаються значення матриці даних для перших двох ознак.

Після натискання на кнопку *Start* починає роботу відповідний алгоритм нечіткої кластеризації з параметрами, встановленими за замовчуванням або зміненими користувачем. Результати роботи алгоритму відображаються на графіку. Знайдені центри кластерів зображені чорними кругами, і їх координати можна зберегти в зовнішньому файлі з метою наступного аналізу.

Іноді число нечітких кластерів, необхідних для роботи алгоритму FCM, апріорі є невідомим. У цьому випадку доцільно використовувати реалізований в системі MATLAB так званий алгоритм субтрактивної кластеризації.

У разі відсутності будь-яких апріорних припущень щодо кількості нечітких кластерів у системі MATLAB можна використовувати функцію командного рядка *subclust* або метод субтрактивної нечіткої кластеризації, реалізований в графічному інтерфейсі кластеризації.

Ідея методу субтрактивної кластеризації полягає в тому, що кожна точка даних передбачається в якості центру потенціального кластера, після чого слід обчислити деяку міру здатності кожної точки даних представляти центр кластера. Ця кількісна міра базується на оцінці щільності точок даних навкруги відповідного центру кластера.

КОНТРОЛЬНІ ЗАПИТАННЯ ТА ЗАВДАННЯ

1. Дайте визначення поняттю кластерний аналіз.
2. Характеристика типів шкал вимірювань.
3. Дайте визначення нечіткому кластерному аналізу.
4. Характеристика нечітких кластерів як нечітких множин.
5. Дайте визначення нечіткому кластерному аналізу.
6. Характеристика типових представників або центрів v_k шуканих нечітких кластерів.
7. Цільова функція нечіткої кластеризації.
8. Переваги постановки задачі нечіткої кластеризації.

ЛІТЕРАТУРА

[12] стор. 383–398; [23] стор. 20–90

РОЗДІЛ 7

ШТУЧНІ НЕЙРОННІ МЕРЕЖІ ТА ЇХ МОДЕЛІ

7.1 Поняття штучної нейронної мережі і основні способи її відтворення

Основним компонентом розглянутих раніше засобів системи MATLAB у рамках пакета Fuzzy Logic Toolbox є база нечітких правил, яка займає центральне місце в процедурах нечіткого висновку. У той же час існують цілі класи прикладних задач, у яких виявлення та побудова нечітких правил неможлива або пов'язана із серйозними труднощами концептуального характеру. До таких задач відносяться завдання розпізнавання образів, екстраполяції та інтерполяції функціональних залежностей, класифікації і прогнозування, оптимізації, нелінійного та ситуаційного керування, а також інтелектуального аналізу даних (Data Mining).

Загальною особливістю подібних завдань є існування деякої залежності або відносини, що зв'язує вхідні й вихідні змінні моделі системи, що представляється у формі так званого «чорного ящика». У цьому разі виявлення й визначення даної залежності в явному теоретико-множинному або аналітичному виді не є можливим або через недолік інформації про модельовану проблемну область, або складності обліку різноманіття факторів, що впливають на характер цього взаємозв'язку.

Для конструктивного рішення подібних завдань розроблений спеціальний математичний апарат, що одержав назву *нейронних мереж*. Початково теорія штучних нейронних мереж була розроблена для моделювання здатності розпізнавання образів людиною.

Перевагою моделей, побудованих на основі нейронних мереж, є можливість одержання нової інформації про проблемну область у формі деякого прогнозу. У цьому разі побудова та налаштування нейронних мереж здійснюється за допомогою їхнього навчання на основі наявної і доступної інформації.

Недоліком нейронних мереж є представлення знань про проблемну область у спеціальному виді, яке може суттєво відрізнятися від можливої змістовної інтерпретації існуючих взаємозв'язків і відносин.

Нечіткі нейронні мережі, або гібридні за задумом їх розроблювачів покликано об'єднати в собі переваги нейронних мереж і систем нечіткого висновку. З одного боку, вони дають змогу розробляти й представляти моделі систем у формі правил нечітких продукцій, які мають наочність і простоту змістовної інтерпретації. З іншого боку, для побудови нечітких правил використовуються методи нейронних мереж, що є більш зручним і менш

трудомістким процесом для системних аналітиків. Останнім часом апарат гібридних мереж повсюди визнається фахівцями як один з найбільш перспективних для рішення слабо або погано структурованих завдань прикладного системного аналізу.

Проблема машинної імітації людських думок надихає вчених вже кілька століть. Понад 50 років тому були створені перші електронні моделі нервових клітин. Крім того, з'являлося багато робіт за новими математичними моделями і навчальними алгоритмами. Сьогодні так звані *нейронні мережі* становлять найбільший інтерес в цій галузі. Вони використовують множину простих обчислювальних елементів, які називаються *нейронами*, кожен з яких імітує поведінку окремої клітини людського мозку. Прийнято вважати, що людський мозок – це природна нейронна мережа, а модель мозку – це штучна нейронна мережа. На рис. 7.1 показана базова структура такої нейронної мережі.

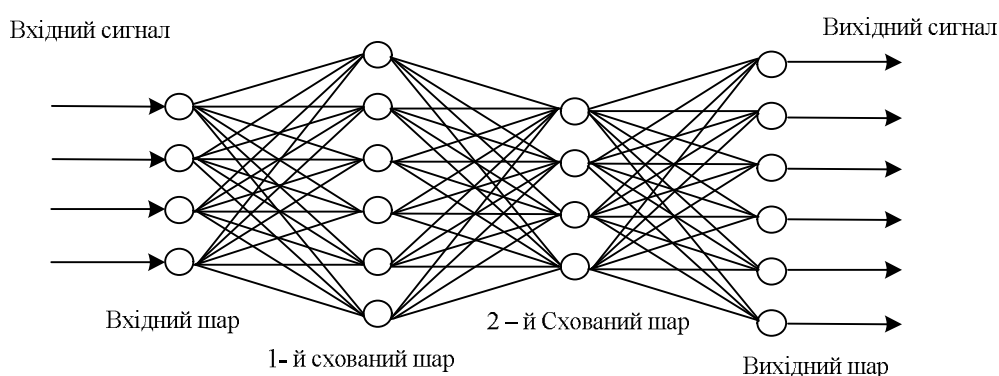


Рис. 7.1. Базова структура штучної нейронної мережі

Кожен нейрон у нейронній мережі здійснює перетворення вхідних сигналів у вихідний сигнал і пов'язаний з іншими нейронами. Вхідні нейрони формують так званий інтерфейс нейронної мережі. Нейронна мережа, показана на рис. 7.1, має шар, який сприймає вхідні сигнали, і шар, що генерує вихідні сигнали. Інформація вводиться в нейронну мережу через вхідний шар. Всі прошарки нейронної мережі обробляють ці сигнали доти, поки вони не досягнуть вихідного шару.

Задача нейронної мережі – виконання перетворення вхідної інформації потрібним чином. Для цього мережа попередньо навчається. При навчанні використовуються ідеальні (еталонні) значення пар «входи–виходи» або «експерт», який оцінює поведінку нейронної мережі. Для навчання використовується так званий навчальний алгоритм. Неналаштована нейронна мережа не здатна відображати бажаної поведінки. Навчальний алгоритм модифікує окремі нейронні ваги її зв'язків так, щоб поведінка мережі відповідала бажаній поведінці.

Дослідниками нейронних мереж запропоновано більше 200 моделей клітин людського мозку. Розглянемо лише ті з них, які найбільш часто використовуються в промислових цілях.

Людський мозок складається з більш ніж 10^{11} нервових клітин, що мають більш 10^{14} взаємозв'язків. На рис. 7.2 показана спрощена схема такого біологічного нейрона. Сама по собі клітина складається з ядра і зовнішньої електромембрани. Кожен нейрон має рівень активації, що лежить в діапазоні між максимумом і мінімумом, отже, на відміну від булевої логіки, існує більш ніж два рівня активації.

Для збільшення або зменшення активності даного нейрона іншими нейронами існують так звані *синапси*. Вони переносять величину активності від нейрона-відправника до нейрона-одержувача. Якщо синапс є збудливим, то величина активності нейрона-відправника збільшує активність нейрона-одержувача. Якщо синапс є гальмівним, то величина активності нейрона-відправника зменшує активність нейрона-одержувача. Синапси відрізняються не тільки за ознакою гальмування або збудження нейрона-одержувача, але також і за сумарним впливом (синаптична потужність). Вихідний сигнал кожного нейрона передається по так званому *аксону*, який закінчується більш ніж 10000 синапсами, що впливають на інші нейрони.

Розглянута модель нейрона лежить в основі більшості сьогоденних застосувань нейронної мережі. Але така модель є лише дуже грубим наближенням дійсності. Насправді ми не можемо змоделювати навіть один єдиний людський нейрон; це понад людських можливостей у моделюванні. Отже, будь – яка робота, що базується на цій простій моделі нейрона, не здатна точно імітувати людський мозок. Однак багато успішних застосувань, що використовують цей метод, забезпечили успіх нейронним мережам, що базуються на простій моделі нейрона.

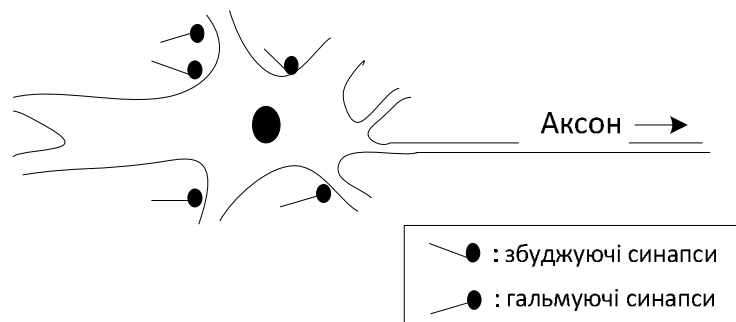


Рис. 7.2. Спрощена схема біологічного нейрона

7.2 Математична модель штучного нейрона

На базі простої концепції побудови нейрона може бути відтворена множина математичних моделей нейрона. На рис. 7.3 показана найбільш загальна схема. Так звана підсумовуюча функція об'єднує всі вхідні сигнали X_i , які надходять від нейронів-відправників. Значенням такого об'єднання є зважена сума, де ваги W_i є синаптичними потужностями. Збуджуючі синапси

мають позитивні ваги, а гальмуючі синапси – негативні ваги. Для вираження нижнього рівня активації нейрона до зваженої суми додається компенсація b (зміщення).

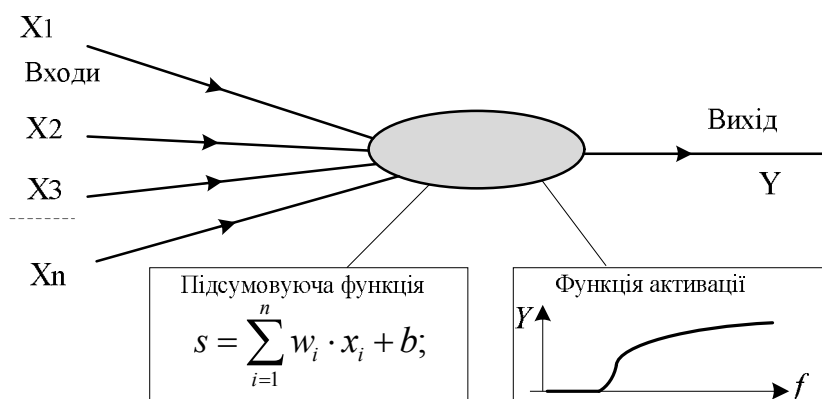


Рис. 7.3. Функціональна модель нейрона

Так звана функція активації розраховує вихідний сигнал нейрона Y за рівнем активності S .

На сьогодні запропонована та вивчена велика кількість варіантів і різновидів нейронних мереж. Для розгляду моделей гібридних мереж необхідно дати деяке загальне визначення нейронної мережі (більш точно – штучної нейронної мережі), яке послужить основою для наступного визначення моделі нейрон–нечіткого висновку.

Концептуальною основою і складовою частиною штучних нейронних мереж є так званий *штучний нейрон*, який має певну внутрішню структуру (рис. 7.4) і правила перетворення сигналів.

Штучний нейрон складається з множників (*синапсів*), суматора і нелінійного перетворювача. *Синапси*, що зображені перекресленим кружком, призначені для зв'язку нейронів між собою і зв'язують вхідний та вихідний сигнали через коефіцієнт, тобто вхідний сигнал x_i домножується на деяке постійне число. Це число w_i , що називаємо *вагою синапсу*, характеризує силу цього зв'язку.

Суматор виконує додавання всіх сигналів, що надходять на вхід нейрона, від інших нейронів і зовнішніх вхідних сигналів.

Нелінійний перетворювач призначений для нелінійної зміни вихідного значення суматора згідно з деякою функцією від одного аргументу. Ця функція називається *функцією активації*, або *передатною функцією* нейрона.

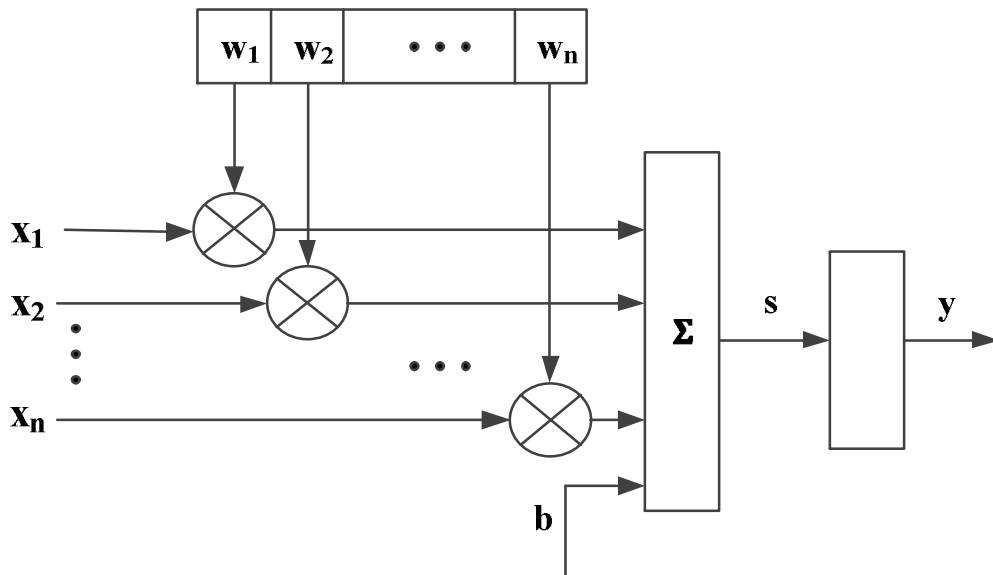


Рис. 7.4. Структура штучного нейрона

Правила перетворення сигналів визначаються математичною моделлю нейрона, яка може бути записана у формі таких аналітичних виразів:

$$s = \sum_{i=1}^n w_i \cdot x_i + b;$$

$$y = f(s),$$

де w_i – вага синапсу ($i \in \{1, 2, \dots, n\}$); b – значення зсуву; s – результат підсумовування; x_i – компонент вектора входу чи вхідного сигналу ($i \in \{1, 2, \dots, n\}$); y – вихідний сигнал нейрона; n – число входів нейрона; f – функція активації (передатна функція) нейрона нелінійне перетворення, що є деяким нелінійним перетворенням. У загальному випадку: $w_i, x_i, b \in R (i \in \{1, 2, \dots, n\})$.

Синаптичні зв'язки з позитивними вагами $w_i \in R^+$ ($i \in \{1, 2, \dots, n\}$) називаються *збуджуючими*, а з негативними вагами $w_i \in R^-$ ($i \in \{1, 2, \dots, n\}$) – *гальмуючими*.

Окремо взятий штучний нейрон повністю описується своєю математичною моделлю (рис. 7.5) у вигляді наведених рівнянь. Одержавши вектор вхідного сигналу x , нейрон видає деяке число y на своєму виході. У якості функції активації нейрона можуть бути використані різні нелінійні перетворення (табл. 7.1).

Нейронна мережа є сукупністю окремих нейронів, взаємозалежних між собою деяким фіксованим чином. У цьому разі взаємозв'язок нейронів визначається або задається структурою (*топологією*) нейронної мережі. З погляду топології нейронні мережі можуть бути повністю зв'язаними,

багатошаровими та слабо зв'язаними. У загальному випадку *структура* багатошарової або багаторівневої нейронної мережі може бути зображена в такий спосіб (рис. 7.6).

Напрямок зв'язку від одного нейрона до іншого є важливим аспектом нейромережі. У більшості мереж кожен нейрон прихованого прошарку отримує сигнали від всіх нейронів попереднього прошарку та звичайно від нейронів вхідного прошарку. Після виконання операцій над сигналами нейрон передає свій вихід до всіх нейронів наступних прошарків, забезпечуючи шлях передачі вперед (*feedforward*) на вихід.

Вихідний прошарок пересилає інформацію безпосередньо до зовнішнього середовища, до вторинного комп'ютерного процесу або до інших пристроїв.

Між цими двома прошарками може бути багато прихованих прошарків, які містять багато нейронів у різноманітних зв'язаних структурах. Входи та виходи кожного з прихованих нейронів просто йдуть до інших нейронів.

Кожний з рівнів нейронної мережі називається її *шаром*. У цьому разі шар вхідного рівня називається *вхідним шаром*, шар рівня 1 і 2 – *схованими шарами*, а шар рівня 3 – *вихідним шаром*.

У свою чергу багатошарові нейронні мережі можуть бути таких типів:

1. Монотонні – кожен шар (крім вхідного) додатково розбивається на 2 блоки: збуджений і гальмуючий. Аналогічно розбиваються й зв'язки між блоками: на збуджені й гальмуючі. У цьому разі в якості функцій активації можуть бути використані тільки монотонні функції (табл. 7.1);
2. Нейронні мережі зі зворотними зв'язками – інформація з наступних шарів може передаватися на нейрони попередніх шарів;
3. Нейронні мережі без зворотних зв'язків – інформація з наступних шарів не може передаватися на нейрони попередніх шарів. Класичним варіантом багатошарової нейронної мережі є повністю зв'язана мережа прямого поширення (рис. 7.5).

Алгоритм створення нейромережових моделей складається з таких етапів:

1. Вибір типу і структури нейронної мережі для рішення поставленої проблеми (синтез структури нейронної мережі).
2. Навчання нейронної мережі (визначення числових значень ваг кожного з нейронів) на основі наявної інформації про рішення цього завдання експертом або даних про рішення завдання в минулому.
3. Перевірка нейронної мережі на основі використання деякого контрольного прикладу (необов'язковий етап).
4. Використання навченої нейронної мережі для рішення поставленої проблеми.

Табл. 7. 1. Основні види функцій активації нейронів

Назва	Формула	Область значень
Лінійна	$f(s) = k \cdot s$	R
Напівлінійна	$f(s) = \begin{cases} 0, & \text{якщо } s \leq 0 \\ k \cdot s, & \text{якщо } s > 0 \end{cases}$	R_+
Гранична	$f(s) = \begin{cases} 0, & \text{якщо } s < T \\ 1, & \text{якщо } s \geq T \end{cases}$	$\{0, 1\}$
Модульна	$f(s) = s $	R_+
Знакова (сигнатурна)	$f(s) = \begin{cases} -1, & \text{якщо } s \leq T \\ 1, & \text{якщо } s > T \end{cases}$	$\{-1, 1\}$
Квадратична	$f(s) = s^2$	R_+
Експонентна	$f(s) = e^{-as}$	R_+
Синусоїдальна	$f(s) = \sin(s)$	$[-1, 1]$
Логістична (сигмоїдальна)	$f(s) = \frac{1}{1 + e^{-as}}$	$(0, 1)$
Раціональна (сигмоїдальна)	$f(s) = \frac{s}{a + s }$	$(-1, 1)$
Гіперболічний тангенс (сигмовидна)	$f(s) = \frac{e^{as} - e^{-as}}{e^{as} + e^{-as}}$	$(-1, 1)$
Лінійна з насиченням (крокова)	$f(s) = \begin{cases} -1, & \text{якщо } s \leq -1 \\ s, & \text{якщо } -1 < s < 1 \\ 1, & \text{якщо } s \geq 1 \end{cases}$	$[-1, 1]$
Напівлінійна з насиченням	$f(s) = \begin{cases} 0, & \text{якщо } s \leq 0 \\ s, & \text{якщо } 0 < s < 1 \\ 1, & \text{якщо } s \geq 1 \end{cases}$	$[0, 1]$
Трикутна	$f(s) = \begin{cases} 1 - s , & \text{якщо } s \leq 1 \\ 0, & \text{якщо } s > 1 \end{cases}$	$[0, 1]$
Радіальна базисна (Гаусова)	$f(s) = e^{-s^2}$	$(0, 1]$

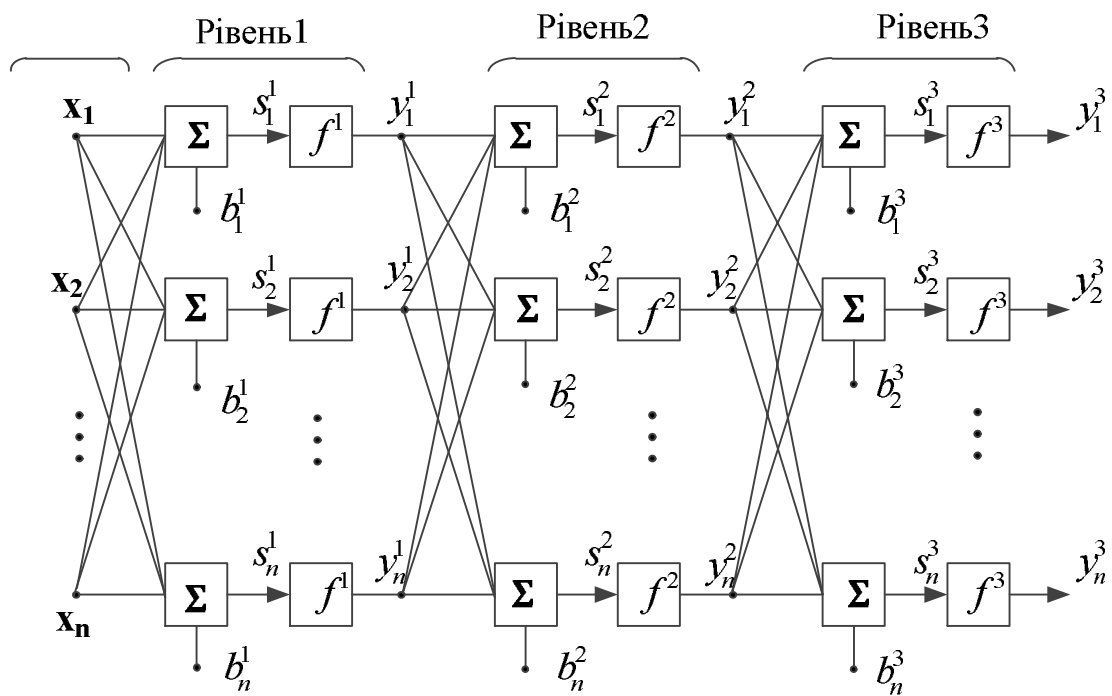


Рис. 7.5. Модель багаторівневої нейронної мережі

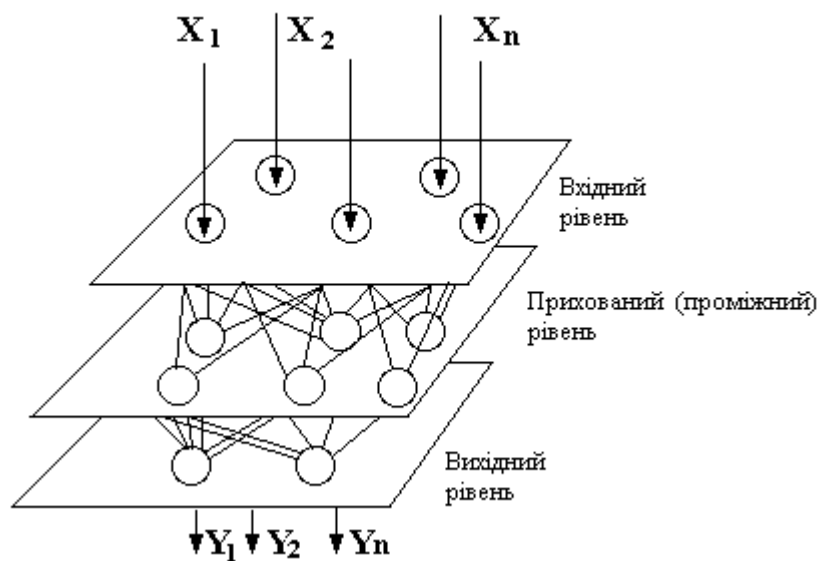


Рис. 7.6. Структура багаторівневої (трирівневої, тришарової) нейронної мережі

7.3 Навчання штучної нейронної мережі

На сьогодні запропоновані різні схеми класифікації нейронних мереж і відповідні алгоритми їх навчання.

Основним моментом у разі проектування нейронної мережі є відтворення її поведінки відповідно алгоритму управління. Це – *фаза навчання*. Для цього використовується так звана *навчальна вибірка* або *експерт*. *Експерт* – це або

математична функція, або особа, яка оцінює якість поведінки нейронної мережі. Оскільки нейронні мережі в основному використовуються в складних застосуваннях, де немає хороших математичних моделей, то навчання проводиться за допомогою навчальної вибірки, тобто еталонних пар «входи–виходи».

Після завершення навчання нейронна мережа готова до використання. Це – *робоча фаза*. В результаті навчання нейронна мережа буде обчислювати вихідні сигнали, близькі до еталонних даних у разі відповідних вхідних сигналів. За проміжних вхідних сигналів мережа апроксимує необхідні вихідні величини. Поведінка нейронної мережі в робочій фазі детермінована, тобто для кожної комбінації вхідних сигналів на виході завжди будуть одні і ті ж сигнали. Протягом робочої фази нейронна мережа не навчається. Це дуже важливо для більшості технічних застосувань, оскільки система не буде прагнути до екстремальної поведінки.

Одним з найпоширеніших алгоритмів навчання є так званий алгоритм *зворотного поширення помилки* (*back propagation*). Цей алгоритм є ітеративним градієнтним алгоритмом мінімізації середньоквадратичного відхилення значень виходу від бажаних значень (мінімізації помилки) у багатошарових нейронних мережах.

Загальна схема навчання зі зворотним поширенням помилок має такий вигляд:

1. *Визначається черговий навчальний вектор і подється на вхід мережі;*
2. *Розраховується вихід мережі;*
3. *Знаходиться відхилення (різниця) між фактичним і бажаним значенням параметра виходу;*
4. *Корегуються вагові коефіцієнти так, щоб мінімізувати помилку;*
5. *Процедура повторюється, поки помилка не досягне прийнятного значення.*

Цей метод навчання багатошарової нейронної мережі називається узагальненим дельта–правилом або правилом *error back propagation* (зворотного поширення помилки). Метод був запропонований в 1986 р. Румельхартом, Макклеландом і Вільямсом. Це ознаменувало відродження інтересу до нейронних мереж, який став згасати на початку 70–х років. Пізніше було виявлено, що Паркер опублікував подібні результати в 1982 р., а Вербос виконав таку роботу в 1984 р. Однак стаття Руммельхарта та ін., опублікована в журналі *Nature* (1986), є досі найбільш цитованою в цій галузі.

Навчання мережі починається з пред'явлення образу і обчислення відповідної реакції. Порівняння з бажаною реакцією дає можливість змінювати ваги зв'язків так, щоб мережа на наступному кроці могла видавати більш точний результат. Навчальне правило забезпечує налаштування ваг зв'язків. Інформація про виходи мережі є вихідною для нейронів попередніх шарів. Ці

нейрони можуть налаштовувати ваги своїх зв'язків для зменшення похибки на наступному кроці.

Коли ми пред'являємо неналаштованій мережі вхідний образ, вона буде видавати деякий випадковий вихід. Функція помилки є різницею між поточним виходом мережі та ідеальним виходом, який необхідно отримати. Для успішного навчання мережі потрібно наблизити вихід мережі до бажаного виходу, тобто послідовно зменшувати величину функції помилки. Це досягається налаштуванням міжнейронних зв'язків. *Узагальнене дельта-правило навчає мережу шляхом обчислення функції помилки для заданого входу з подальшим її зворотним поширенням (ось звідки назва) від кожного шару до попереднього.* Кожен нейрон у мережі має свої ваги, які налаштовуються, щоб зменшити величину функції помилки. Для нейронів вихідного шару відомі їхні фактичні та бажані значення виходів. Тому налаштування ваг зв'язків для таких нейронів є відносно простим. Однак для нейронів попередніх шарів налаштування не настільки очевидне. Інтуїтивно ясно, що нейрони внутрішніх шарів, які пов'язані з виходами, що мають велику похибку, мають змінювати свої ваги значно сильніше, ніж нейрони, з'єднані з майже коректними виходами. Іншими словами, *ваги такого нейрона мають змінюватися прямо пропорційно помилці тих нейронів, з якими цей нейрон пов'язаний.* Ось чому зворотне поширення цих помилок через мережу дає змогу коректно налаштовувати ваги зв'язків між усіма шарами. У цьому випадку величина функції помилки зменшується і мережа навчається.

Основні співвідношення методу зворотного поширення помилки отримані за таких позначень:

- E_p – величина функції помилки для образу p ;
- t_{pj} – бажаний вихід нейрона j для образу p ;
- o_{pj} – дійсний вихід нейрона j для образу p ;
- w_{pj} – вага зв'язку між i – і j – м нейронами.

Перевага використання сигмоїдної функції як нелінійного елемента полягає в тому, що вона дуже нагадує крокову функцію і таким чином може демонструвати поведінку, подібну природному нейрону. Сигмоїдна функція визначається як

$$f(net) = \frac{1}{1 + e^{-k \cdot net}}$$

і має діапазон $0 < f(net) < 1$, де net – вагова сума активації кожного нейрона; k – позитивна константа, що впливає на розтягнення функції: збільшення k стискає функцію, а при досягненні $k \rightarrow \infty$ функція $f(net)$ наближається до функції Хевісайда². Коефіцієнт k може використовуватися як параметр посилення, оскільки для слабких вхідних сигналів кут нахилу буде досить крутим і

² Кусково-постійна функція, яка дорівнює нулю у разі від'ємних значень аргумента і одиниці – у разі позитивних. У нулі вона не визначена, але її звичайно в цій точці представляють деяким числом, що б область визначення вміщувала всі точки дійсної осі.

відповідно посилення будуть набагато меншими. Це означає, що мережа може приймати великі сигнали і у цьому разі залишатися чутливою до слабких змін сигналу.

Проте головний сенс у використанні цієї функції полягає в тому, що вона має просту похідну, і це значно полегшує застосування *back propagation*-методу. Якщо вихід нейрона O_{pj} задається як

$$o_{pj} = f(net) = \frac{1}{1 + e^{-k \cdot net}},$$

то похідна по відношенню до цього нейрона $f'(net)$ обчислюється так:

$$f'(net) = \frac{ke^{-k \cdot net}}{(1 + e^{-k \cdot net})^2} = \frac{kf(net)}{1 - f(net)} = ko_{pj}(1 - o_{pj}),$$

тобто є простою функцією від виходів нейронів.

7.3.1 Алгоритм налаштування нейронної мережі

Наведений алгоритм налаштування багатошарової нейронної мережі з використанням *backpropagation* – правила навчання. Для його застосування необхідно, щоб нейрони мали безперервно диференціюючу нелінійну порогову функцію активації. Нехай це буде сигмоїдна функція з досить простою похідною

$$f'(net) = \frac{1}{1 + e^{-k \cdot net}}.$$

Алгоритм навчання полягає в такому:

1. *Задати початкові значення ваг і порогів кожного нейрона.* Всім вагам і порогам присвоюються малі випадкові значення.
2. *Представити вхідний і вихідний вектор з навчальної вибірки.* Нехай $X_p = \{x_0, x_1, \dots, x_{n-1}\}$ – поточний вхідний вектор, $T_p = \{t_0, t_1, \dots, t_{m-1}\}$ – поточний вихідний вектор з навчальної вибірки, де n – число нейронів вхідного шару, m – число нейронів вихідного шару.
У разі вирішення задач класифікації образ T_p може складатися з нулів, крім одного елемента, що дорівнює 1, який і буде визначати клас поточного вхідного вектора.
3. *Розрахувати дійсні значення виходів.* Значення виходів нейронів кожного шару розраховуються як

$$y_{pj} = f \left[\sum_{i=0}^{n-1} w_i x_i \right]$$

і передаються на входи нейронів наступного шару. Вихідні значення нейронів вихідного шару рівні o_{pj} . У цьому разі $w_0 = -b$ (початковий зсув) і $x_0 = 1$.

4. *Провести модифікацію ваг зв'язків.* Починаючи від вихідного шару і рухаючись в зворотному напрямку, необхідно змінювати ваги зв'язків таким чином:

$$w_{ij}(t+1) = w_{ij}(t) + \eta \delta_{pj} o_{pj},$$

де w_{ij} – вага зв'язку між i - і j -м нейронами на t -му кроці; η – швидкість навчання; δ_{pj} – швидкість зміни помилки для нейрона j у разі пред'явлення вектора p . Для нейрона вихідного шару

$$\delta_{pj} = k o_{pj} (1 - o_{pj}) (t_{pj} - o_{pj}),$$

для нейронів внутрішніх шарів

$$\delta_{pj} = k o_{pj} (1 - o_{pj}) \sum_k \delta_{pk} w_{jk},$$

де під знаком суми стоять величини, що відносяться до нейронів наступного шару.

Вибір виду та структури нейронної мережі визначається специфікою розв'язуваного завдання. У цьому разі для рішення окремих типів практичних завдань розроблені оптимальні конфігурації нейронних мереж, які найбільш адекватно відображають особливості відповідної проблемної області.

Гібридні мережі є подальшим розвитком нейронних мереж, які реалізовані в пакеті Fuzzy Logic Toolbox системи MATLAB.

Електронна реалізація штучного нейрона представлена на рис. 7.7, а електрична схема прототипа нейрона – на рис. 7.8.

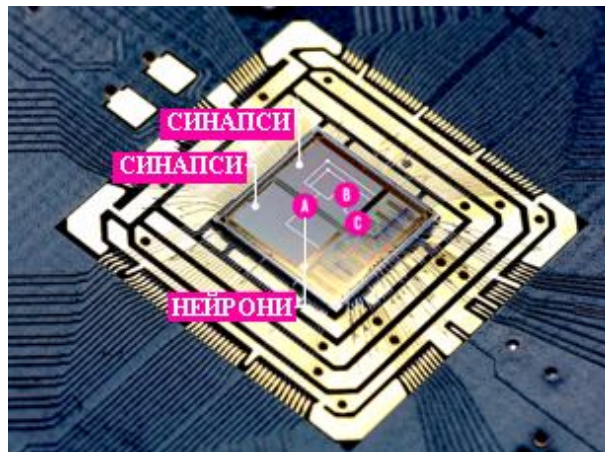


Рис. 7.7. Збільшений вигляд електронної мікросхеми (chip) штучного нейрона (прототип нейрона біологічних систем)³

³ [<http://spectrum.ieee.org/computing/hardware/the-brain-as-computer-bad-at-math-good-at-everything-else>]



Рис. 7.8. Електрична схема прототипу штучного нейрона

7.4 Загальна характеристика програмного інтерфейсу адаптивних систем нейронечіткого висновку

Нейронні мережі в системі MATLAB представлені програмним доповненням Neural Network Toolbox, а гібридні нейронні мережі та засоби їх розробки – програмним пакетом для моделювання адаптивних систем нейронечіткого висновку (ANFIS, Adaptive Neuro–Fuzzy Inference System). Робоче вікно графічного інтерфейсу програми ANFIS наведено на рис.7.9.

Історія досліджень в галузі нейронних мереж бере свій початок з 1943 р., коли У. Маккалох (W. McCulloch) і У. Піттс (W. Pitts) запропонували першу модель нейрона й сформулювали основні принципи функціонування головного мозку людини. Перші результати в теорії нейронних мереж пов'язують із появою в період 1960–1970 рр. різних моделей штучних нейронних мереж, які були запропоновані в роботах В. Уїдроу (W. Widrow), К. Штайнбуха (K. Steinbuch), М. Мінського (M. Minsky) і С. Пейперта (S. Papert). Однак початком сучасного етапу розвитку нейронних мереж прийнято вважати 1982р., коли американським математиком Дж. Хопфілдом (J. Hopfield) був розроблений спеціальний клас нейронних мереж і запропоновані методи їх навчання.

Зростання популярності нейронних мереж пов'язано з появою перших комерційних програмних засобів, що дають змогу здійснювати побудови відповідних моделей для рішення різних прикладних завдань.

Розгляд можливостей пакета Neural Network та особливостей побудови і використання гібридних мереж, реалізованих в пакеті Adaptive Neuro–Fuzzy Inference System та Fuzzy Logic Toolbox, є темою самостійного вивчення.

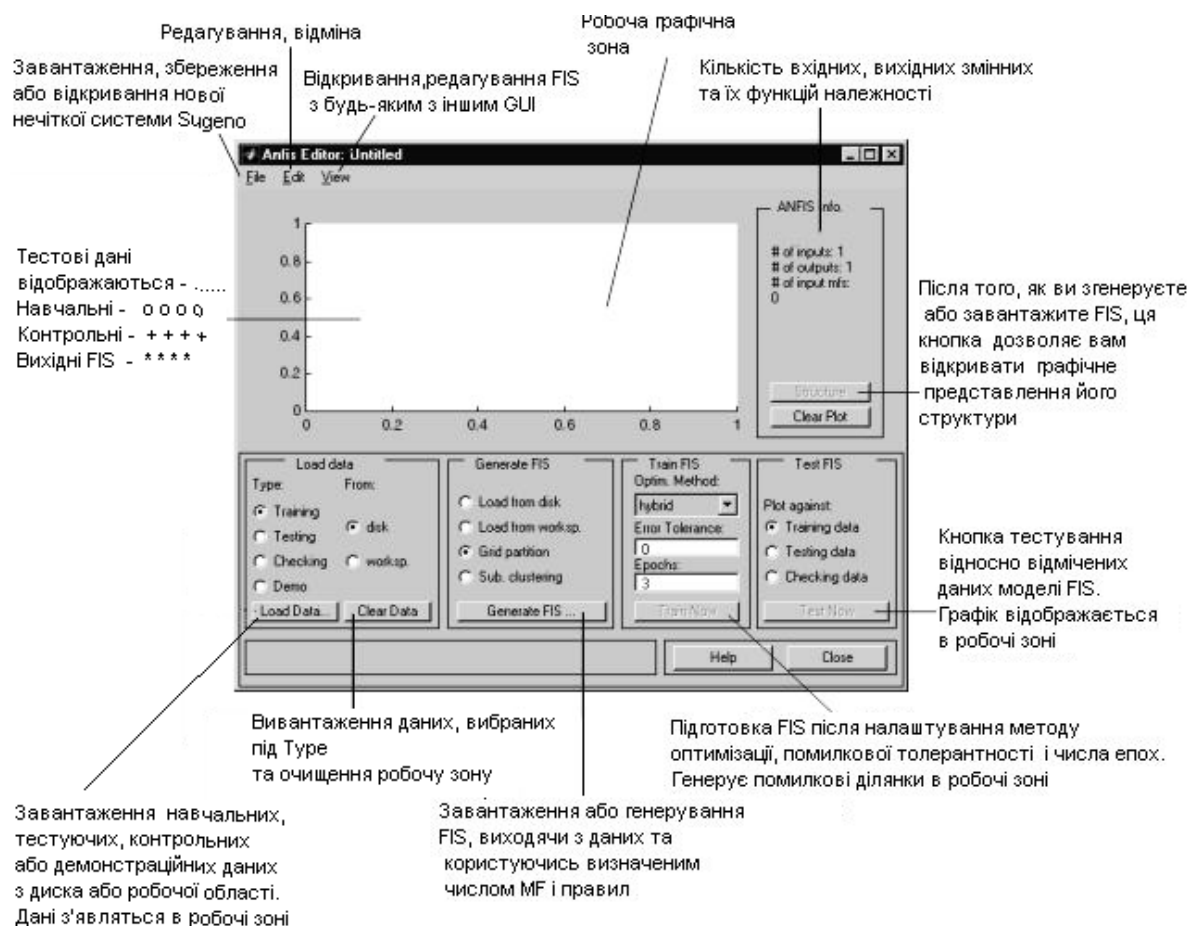


Рис. 7.9. Робоче вікно програмного пакета ANFIS

КОНТРОЛЬНІ ЗАПИТАННЯ ТА ЗАВДАННЯ

1. Дайте визначення нейронної мережі.
2. Дайте характеристику структурі штучного нейрона.
3. Що таке синапси та їх призначення?
4. Дайте характеристику елементам штучного нейрона.
5. Основні види функцій активації нейронів.
6. Характеристика процесу побудови і використання нейромережових моделей.
7. Структура штучного нейрона.
8. Нелінійний перетворювач та його призначення.
9. Алгоритм навчання нейронної мережі.
10. Дайте характеристику алгоритму зворотного поширення помилки.
11. Загальна схема навчання зі зворотним поширенням помилки.
12. Основні співвідношення методу зворотного поширення помилки.
13. Перевага використання сигмоїдної функції як нелінійного елемента.

ЛІТЕРАТУРА

[16], стор. 6–20, стор. 201–229; [12], стор. 426–450; [15], стор. 176–200; [19], стор. 5–50; [25], індекс В;

РОЗДІЛ 8

ОПТИМІЗАЦІЯ ПАРАМЕТРІВ ЕЛЕКТРОЕНЕРГЕТИЧНИХ СИСТЕМ НА ОСНОВІ МАТЕМАТИЧНОГО АПАРАТА ГЕНЕТИЧНИХ АЛГОРИТМІВ

8.1 Генетичний алгоритм та переваги його використання для оптимізаційних задач у порівнянні з класичними методами

Генетичний алгоритм (ГА) (genetic algorithm) – це послідовність математичних операцій, які використовуються для знаходження наближених розв'язків складних важкорозв'язуваних задач (оптимізації, моделювання, створення зразків нової техніки) за допомогою застосування принципів еволюційної біології та інформатики, тобто шляхом послідовного підбору, комбінування і варіації шуканих рішень з використанням механізмів, що нагадують біологічну еволюцію.

Особливістю генетичного алгоритму є акцент на використання оператора «схрещування» (селекції), який виконує операцію рекомбінації рішень–кандидатів, роль якої аналогічна ролі схрещування в живій природі. Засновником генетичних алгоритмів вважається Джон Голланд (*John Holland*), книга якого «Адаптація в природних і штучних системах» (*Adaptation in Natural and Artificial Systems*) є фундаментальною в цій сфері досліджень.

Пошук оптимальних параметрів системи є одним з найбільш універсальних методів знаходження рішення для випадків, коли апріорі не відома послідовність кроків, що ведуть до оптимуму.

Використовуються дві пошукові стратегії: *експлуатація найкращого рішення і дослідження простору рішень*. До першої відноситься градієнтний метод. Він є прикладом стратегії, що вибирає найкраще рішення для оптимізації, ігноруючи в той же час дослідження всього простору пошуку. До другої стратегії відноситься випадковий пошук, який, навпаки, досліджує простір рішень, ігноруючи дослідження перспективних областей пошукового простору.

Генетичний алгоритм є класом пошукових методів загального призначення, які комбінують елементи обох стратегій. Використання цих методів дає змогу утримувати прийнятний баланс між дослідженням всього простору і експлуатацією найкращого рішення.

Загалом алгоритм розв'язання оптимізаційних задач є послідовністю обчислювальних кроків, які асимптотично призводять до оптимального рішення. Більшість класичних методів оптимізації генерують детерміновану послідовність обчислень, що базуються на градієнті або похідній цільової функції вищого порядку. Ці методи застосовуються до однієї вихідної точки

пошукового простору. Потім рішення поступово поліпшується в напрямку найшвидшого зростання або зниження значення цільової функції. У разі такого підходу від «точки до точки» існує небезпека потрапляння в локальний оптимум.

Генетичний алгоритм може здійснювати одночасний пошук за багатьма критеріями (напрямами) шляхом використання популяції можливих рішень. Перехід від однієї популяції до іншої дає змогу уникнути попадання в локальний оптимум. Популяція зазнає щось на зразок еволюції: в кожному поколінні відносно хороші рішення репродукуються, тоді як відносно погані відмирають. ГА використовують імовірнісні правила для визначення хромосоми, що репродукується, або хромосоми, що знищуються, щоб направити пошук до областей ймовірного поліпшення цільової функції.

Можна виділити дві головні переваги генетичних алгоритмів перед класичними оптимізаційними методами:

1. ГА не має значних математичних вимог до видів цільових функцій і їх обмежень. Дослідник не має спрощувати модель об'єкта, втрачаючи її адекватність, і штучно добиваючись можливості застосування доступних математичних методів. У ГА можуть використовуватися найрізноманітніші цільові функції і види обмежень (лінійні і нелінійні), певні на дискретних, безперервних і змішаних універсальних множинах.
2. Еволюційні операції генетичних алгоритмів дають змогу ефективно знаходити глобальний оптимум. За використання класичних покрокових методів глобальний оптимум може бути знайдений тільки в тому випадку, коли функція має властивість опуклості.

Вхідна інформація для виконання генетичного алгоритму кодується таким чином, щоб вирішення задачі могло бути представлено у вигляді масиву, подібного до інформації складу хромосоми. Цей масив часто називають саме як «хромосома». Випадковим чином в масиві створюється деяка кількість початкових елементів «осіб», або початкова популяція. Особи оцінюються з використанням функції пристосованості (цільової або фітнес-функції), в результаті якої кожній особі присвоюється певне значення пристосованості, яке визначає можливість виживання особи. Цільова функція не обов'язково має бути математичною формулою. Вона може бути логічним правилом чи якісним показником.

Генетичні алгоритми в своїх моделях використовують еволюційні принципи спадковості, мінливості і природного відбору. Найбільш поширені три моделі ГА:

1. Класична (канонічна) модель ГА;
2. Генітор (Genitor);
3. Гібридна модель ГА.

У класичному генетичному алгоритмі за допомогою пропорційного відбору формується проміжний масив, з якого випадковим чином вибираються

двоє батьків. Далі використовується одноточковий кросовер, і створені два нащадка мутують (одноточкова мутація) із заданою вірогідністю. Після мутації нащадки займають місця своїх батьків. Процес продовжується доти, поки не буде досягнутий критерій закінчення алгоритму. Модель класичного ГА має такі характеристики: *фіксований розмір популяції; фіксована розрядність генів; пропорційний відбір; особини для схрещування вибираються випадковим чином; одноточковий кросовер і одноточкова мутація; наступне покоління формується з нащадків поточного покоління без «елітизму», нащадки займають місця своїх батьків.*

Для реалізації моделі «генітор» використовується специфічна стратегія відбору. Спочатку популяція ініціюється і її особи оцінюються. Потім вибираються випадковим чином дві особи, схрещуються, внаслідок чого виходить тільки один нащадок, який оцінюється і займає місце найменш пристосованої особи. Далі, необхідно знову випадковим чином вибрати дві особи, схрестити і їх нащадок займає місце особи з найнижчою пристосованістю. Так на кожному кроці в популяції оновлюється тільки одна особа. Така модель ГА має наступні характерні особливості: *фіксований розмір популяції; фіксована розрядність генів; особини для схрещування вибираються випадковим чином; обмежень на тип кросовера і мутації немає; внаслідок схрещування особин виходить один нащадок, який займає місце найменш пристосованої особи.*

Використання *гібридного алгоритму* дає змогу об'єднати переваги ГА з перевагами класичних методів. Річ у тім, що ГА є робастними алгоритмами, тобто вони дають змогу знаходити оптимальне рішення, але знаходження його найчастіше є набагато більш ітераційно–важким завданням з причини стохастичності принципів роботи алгоритму. Внаслідок цього для зменшення часу обчислень ГА використовуються на початковому етапі для ефективного звуження простору пошуку навколо глобального екстремуму, а потім, визначивши кращу особу (набір даних), застосувати один з «класичних» методів оптимізації, підвищивши таким чином ефективність розрахунку в цілому. Після аналізу наведеної моделі ГА можна виділити такі характерні властивості гібридного алгоритму: *фіксований розмір популяції; фіксована розрядність генів; будь–які комбінації стратегій відбору та формування наступного покоління; обмежень на тип кросовера і мутації немає; використання ГА на початковій стадії вирішення задач оптимізації, а потім в роботу включаються класичні методи оптимізації.*

Необхідно зазначити, що в «генітор» моделі пошук оптимальних рішень відбувається набагато краще, ніж в інших моделях, а також збіжність пошуку швидша, ніж у класичного ГА.

Генетичні алгоритми можна використовувати для вирішення таких задач в електроенергетиці:

1. Оптимізація функціонування;

2. Оптимізація запитів у базах даних;
3. Різноманітні задачі на графах (топологічних схемах);
4. Налаштування і навчання штучної нейронної мережі;
5. Задачі визначення стану обладнання;
6. Створення розкладів ремонтних та регламентних робіт;
7. Оптимізація стратегії розвитку енергосистем;
8. Апроксимація функцій;
9. Формування систем штучного інтелекту для прийняття рішень щодо управління;
10. Прогнозування електроспоживання та оптимізація графіків навантаження;
11. Оптимізація процесів енергоринку.

8.2 Послідовність математичних операцій генетичного алгоритму

Генетичні алгоритми використовуються для знаходження рішень у дуже великих просторах пошуку.

Оптимізація є найважливішим етапом вирішення завдань ідентифікації, в тому числі й в електроенергетиці. Основні труднощі застосування класичних методів оптимізації нелінійних функцій пов'язані з проблемами локального екстремуму (рис. 8.1) і «прокляття розмірності» (рис. 8.2).

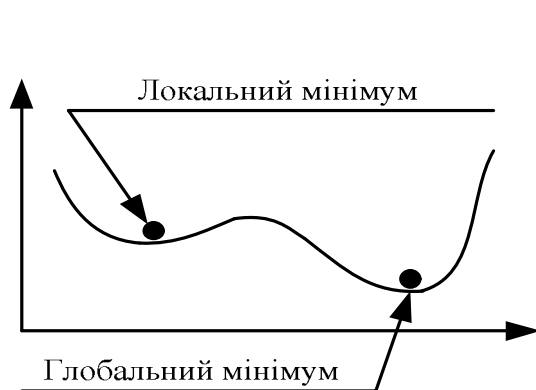


Рис. 8.1. Проблема локального екстремуму

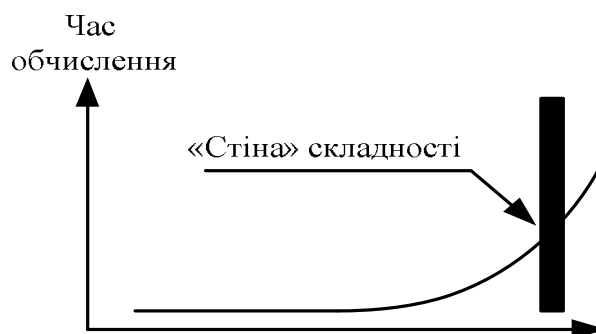


Рис. 8.2. Проблема «прокляття розмірності»

Виконання генетичного алгоритму включає такі етапи:

1. Створення початкової популяції;
2. Обчислення цільової функції пристосованості для осіб популяції (оцінювання);
3. Виконання послідовності операцій генетичного алгоритму до зупинки за певним критерієм за циклом:

- Вибір індивідів із поточної популяції (селекція);
- Схрещування або/та мутація;
- Обчислення функції пристосовуваності для всіх осіб;
- Формування нового покоління.

Генетичний алгоритм можна представити графічно послідовністю основних операцій, зображеною на рис. 8.3.



Рис. 8.3. Генетичний алгоритм

8.2.1 Формування початкової популяції

Спочатку необхідно випадковим чином сформувати деяку початкову популяцію особин. Навіть якщо популяція виявиться абсолютно неконкурентноздатною, генетичний алгоритм все одно досить швидко переведе її в придатну для функціонування (життя) популяцію. Таким чином, на першому кроці можна не намагатися сформувати надто пристосованих осіб (оптимальних рішень), достатньо, щоб вони відповідали формату осіб популяції і на них можна було розрахувати цільову функцію пристосованості (*фітнес функцію*).

Кодування генетичної інформації виконується бінарним бітовим ланцюжком за аналогією з хромосомою. (рис. 8.4).

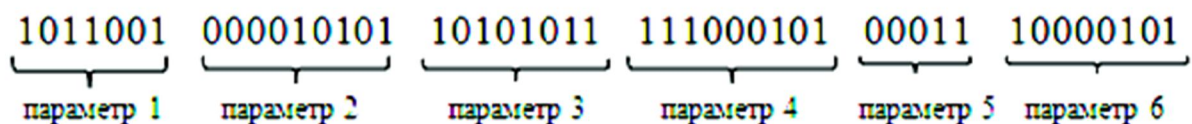


Рис. 8.4. Бінарний бітовий ланцюжок – хромосома з певним фенотипом

Визначення обсягу та складу початкової популяції можливих рішень та заповнення бітового ланцюжка, який відображає загальний притаманний спектру властивостей або параметрів, виконують за допомогою генератора випадкових чисел. У разі наявності апріорної інформації формування фенотипів рішень (вибраних властивостей або параметрів) виконують відповідним чином. Популяція, як правило, не має перевищувати 20...30 осіб.

8.2.2 Селекція (відбір «батьків»)

На етапі відбору необхідно із всієї популяції вибрати її певну частку, яка залишиться дієвою або «живою» на цьому етапі популяції. Є декілька способів провести відбір.

Одним з найбільш ефективних способів формування батьківських пар є метод рулеткового відбору, проілюстрований на рис. 8.5. Кожній особині поточної популяції схем відповідає сектор «рулетки», причому ширина сектора обернено пропорційна значенню цільової функції для відповідного фенотипу. В результаті найкращі технічні рішення, наявні в поточній популяції, можуть дати генетичний матеріал для породження нового покоління (дочірніх) рішень. Водночас найгірші технічні рішення також мають змогу брати участь у селекції, підтримуючи необхідну різноманітність у популяції.

Зазначимо, що існують й інші підходи до формування батьківських пар, наприклад, методом організації турнірів.

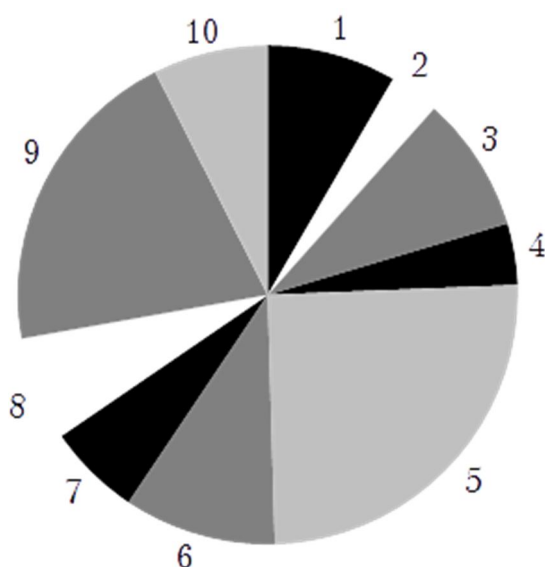


Рис. 8.5. Метод рулеткового відбору

8.2.3 Розмноження (схрещування, кросовер, кросинговер)

Розмноження або схрещування є головною генетичною операцією. Ця операція виконується над двома хромосомами – батьками і створює нащадка шляхом комбінування особливостей (параметрів) обох батьків. Для формування нової особини, тобто нащадка, виберемо деяку випадкову точку в бітовому ланцюжку (точка схрещування – *Cut-point*). Після цього створимо хромосому-нащадок шляхом комбінування сегмента першого батька, який стояв ліворуч від обраної точки схрещування, з сегментом другого з батьків, того, що стоїть праворуч від точки схрещування (рис. 8.6.).

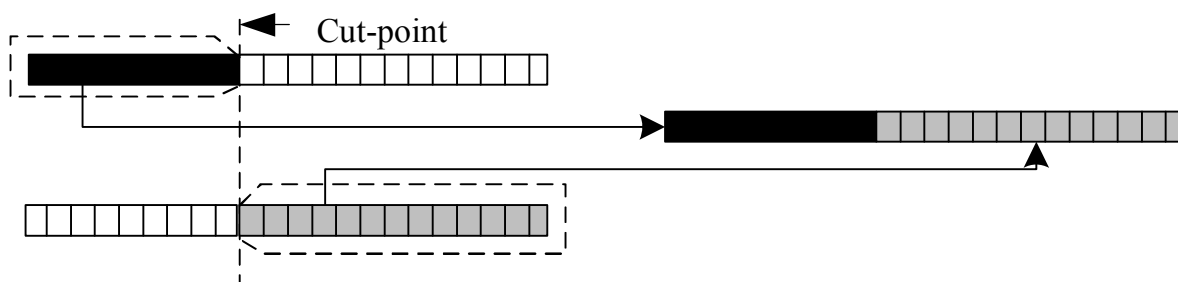


Рис. 8.6. Операція схрещування

Розмноження в генетичних алгоритмах зазвичай статеве – щоб відтворити або «генерувати» нащадка, необхідно декілька батьків, зазвичай потрібна участь двох. Розмноження в різних алгоритмах описується по-різному – воно звісно залежить від формату осіб.

Хромосоми є бітовими ланцюжками. Крім того продуктивність всього генетичного алгоритму в першу чергу залежить від продуктивності операції, що використовується для схрещування.

Частка генерованих на кожній ітерації нащадків називається коефіцієнтом схрещування p_c . Добуток p_c на *розмір_популяції* показує кількість нащадків. Велике значення цього коефіцієнта дає змогу дослідити більше областей простору пошуку (або простору рішень) і зменшує шанс потрапляння в локальний мінімум. Але якщо значення p_c занадто велике, то це призведе до великих витрат часу обчислень на дослідження безперспективних областей.

Розмноження або оператор рекомбінації застосовують відразу ж після оператора відбору батьків для отримання нових особин-нащадків. Сенса рекомбінації полягає в тому, що створені нащадки мають наслідувати генну інформацію від обох батьків. Розрізняють дискретну рекомбінацію і кросинговер (кросовер). На рис. 8.7 наведено графічну ілюстрацію кросинговеру.

Особин для розмноження (батьківські пари) зазвичай вибираються із всієї популяції, а не з тих, що вижили на першому кроці (хоча останній варіант теж має право на існування). Головна проблема генетичних алгоритмів – недостача

різноманітності (*diversity*) в особах. Достатньо швидко виділяється єдиний генотип, який є локальним максимумом і згодом всі елементи популяції програють йому у відборі, і вся популяція «забивається» клонами (копіями) цієї особи чи рішення. Існують різні способи боротьби із таким небажаним ефектом. Один з них – вибір для розмноження не з самих «пристосованих», а взагалі зі всіх осіб.

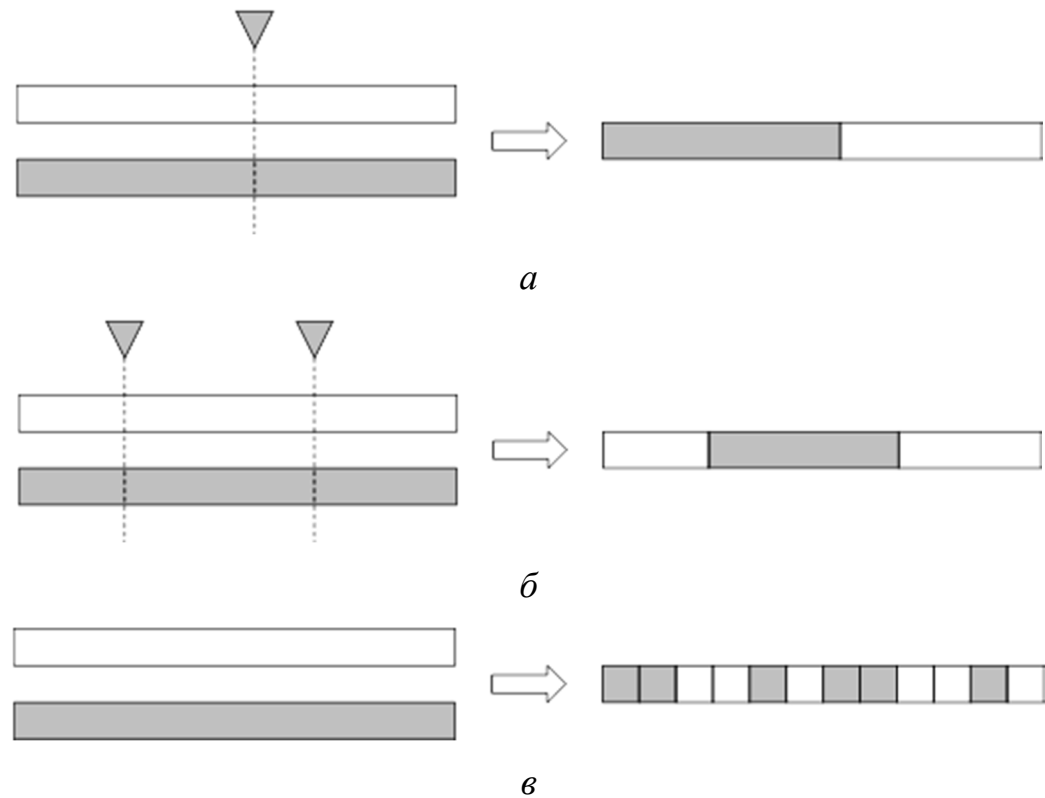


Рис. 8.7. Операція кросинговеру: *а* – одноточковий кросинговер; *б* – двоточковий кросинговер; *в* – багатоточковий кросинговер

Головна вимога до розмноження це те, що нащадок чи нащадки мали можливість успадкувати риси (властивості, параметри, характеристики) всіх батьків шляхом «змішавання» їх якимось достатньо розумним чином.

8.2.4 Мутація

Мутація – це фонові операції, що призводять до випадкових змін у різних хромосомах. Найпростіший варіант мутації складається з випадкової зміни одного або більше генів. В ГА мутація відіграє важливу роль для відновлення генів, що випали з популяції в ході операції вибору, так що вони можуть бути випробувані в нових комбінаціях, та формування генів, які не були представлені у вихідній популяції. Інтенсивність мутацій визначається коефіцієнтом мутації p_m . Він є часткою генів, що піддаються мутації на даній ітерації, в розрахунку на їх загальне число. Занадто мале значення цього

коефіцієнта призводить до того, що багато генів, які могли б бути корисними, ніколи не будуть розглянуті. У той же час занадто велике значення коефіцієнта p_m призведе до великих випадкових збурень. Нащадки перестануть бути схожими на батьків і алгоритм втратить можливість навчатися, зберігаючи спадкові ознаки.

Чинником проведення операції мутації може бути тенденція поведінки цільової функції, виявлена співставленням її з прикладним об'єктом (популяцією рішень) у процесі генерування нових поколінь.

Мутація генетичного матеріалу проілюстрована на рис. 8.8.

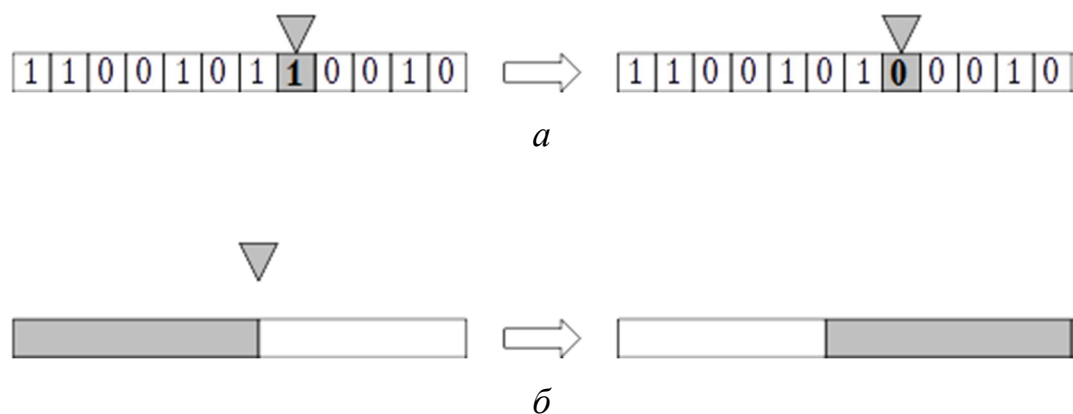


Рис. 8.8. Операція мутації: а – проста мутація; б – інверсія

Роботу генетичного алгоритму завершують, якщо кількість поколінь генерованих дочірніх рішень досягла заданого граничного значення або вичерпано час, який відпущено на еволюцію, або всі фенотипи поточної популяції можливих конфігурацій схем електричної мережі збіглися до одного рішення. Найкращу особину поточної популяції (рішення з найменшим значенням цільової функції) приймають як оптимальну.

У табл. 8.1 наведено основні терміни теорії генетичних алгоритмів.

Табл. 8.1. Основні терміни теорії ГА

Термін ГА	Пояснення
1. Хромосоми	Рішення (код) – бітовий ланцюжок
2. Ген	Частина рішення (декілька біт у хромосомі)
3. Локус	Позиція гена в хромосомі (місцезнаходження)
4. Аллель	Значення гена
5. Фенотип	Розкодоване рішення – рішення з конкретними ознаками (параметрами)
6. Генотип	Закодоване рішення – з набором можливих ознак

8.3 Метод визначення оптимальної конфігурації трифідерної розподільної електричної мережі з використанням генетичного алгоритму

Розглянемо послідовність дій за генетичним алгоритмом щодо знаходження оптимальної конфігурації мережі за функцією оптимальності, яка дає змогу одночасно за чотирма критеріями (мінімальною зв'язаністю схеми, відхиленням напруги, економічною складовою збитків від недовідпуску електроенергії, втратами активної потужності) забезпечити формування оптимальної конфігурації електричної мережі. Функція оптимальності – цільова функція має наступний вигляд:

$$Z = k_1 \sum_{i=1}^N K_i + k_2 \sum_{i=1}^M |\delta U_i| + k_3 \sum_{i=1}^M \alpha_i P_i Y_i + k_4 \sum_{i=1}^L \Delta P_i \rightarrow \min, \quad (8.1)$$

де k_1, k_2, k_3, k_4 – вагові коефіцієнти, призначені для поєднання у складі цільової функції неспіввимірних параметрів; K_i – сигнальний коефіцієнт, який дорівнює одиниці у разі, якщо контакти i -го комутаційного апарата замкнено, і нулю, якщо контакти i -го комутаційного апарата розімкнено; N – загальна кількість комутаційних апаратів, які визначають конфігурацію робочої схеми електричної мережі; δU_i – відхилення робочої напруги i -го пункту схеми за допустимі межі; M – загальна кількість вузлових точок у складі схеми; α_i – ступінь обмеження навантаження i -го пункту робочої схеми у разі виникнення аварійного збурення; P_i – активна потужність навантаження i -го пункту схеми; Y_i – питомі економічні збитки, обумовлені обмеженням навантаження i -го пункту схеми; ΔP_i – втрати активної потужності на i -й ділянці робочої схеми розподільної електричної мережі; L – загальна кількість ділянок у складі робочої схеми.

Метод визначення оптимальної конфігурації трифідерної розподільної електричної мережі реалізується наступними діями та процедур з використанням генетичного алгоритму.

1. Розглядаються послідовно (попарно) всі джерела живлення розподільної мережі, і на основі міні–максного методу виділяють зони безумовного атрактора кожного джерела. Для цього максимізують потужності навантажень підстанцій (пунктів) одного джерела живлення та мінімізують потужності навантажень іншого джерела живлення. Далі знаходять довжину лінії (фідера) l_x до ділянки, якій належить умовний електричний центр, за виразом

$$l_x = \frac{\sum_{i=1}^N P_i l_{0i}}{\sum_{i=1}^N P_i},$$

де N – загальна кількість ділянок лінії з двобічним живленням; P_i – активна потужність навантаження пункту наприкінці i -ї ділянки; l_{0i} – відстань від початку лінії до кінця i -ї ділянки.

Умовне вилучення цієї ділянки поділяє схему на дві незв'язні частини, одна з яких визначає зону безумовного атрактора поточного джерела живлення. Такі операції виконують також для інших джерел живлення. Частина електричної схеми, яка не потрапила до складу жодного безумовного атрактора, утворює «пограничну» зону, в якій буде здійснюватись пошук місць розміщення резервних перемичок (рис. 8.9).

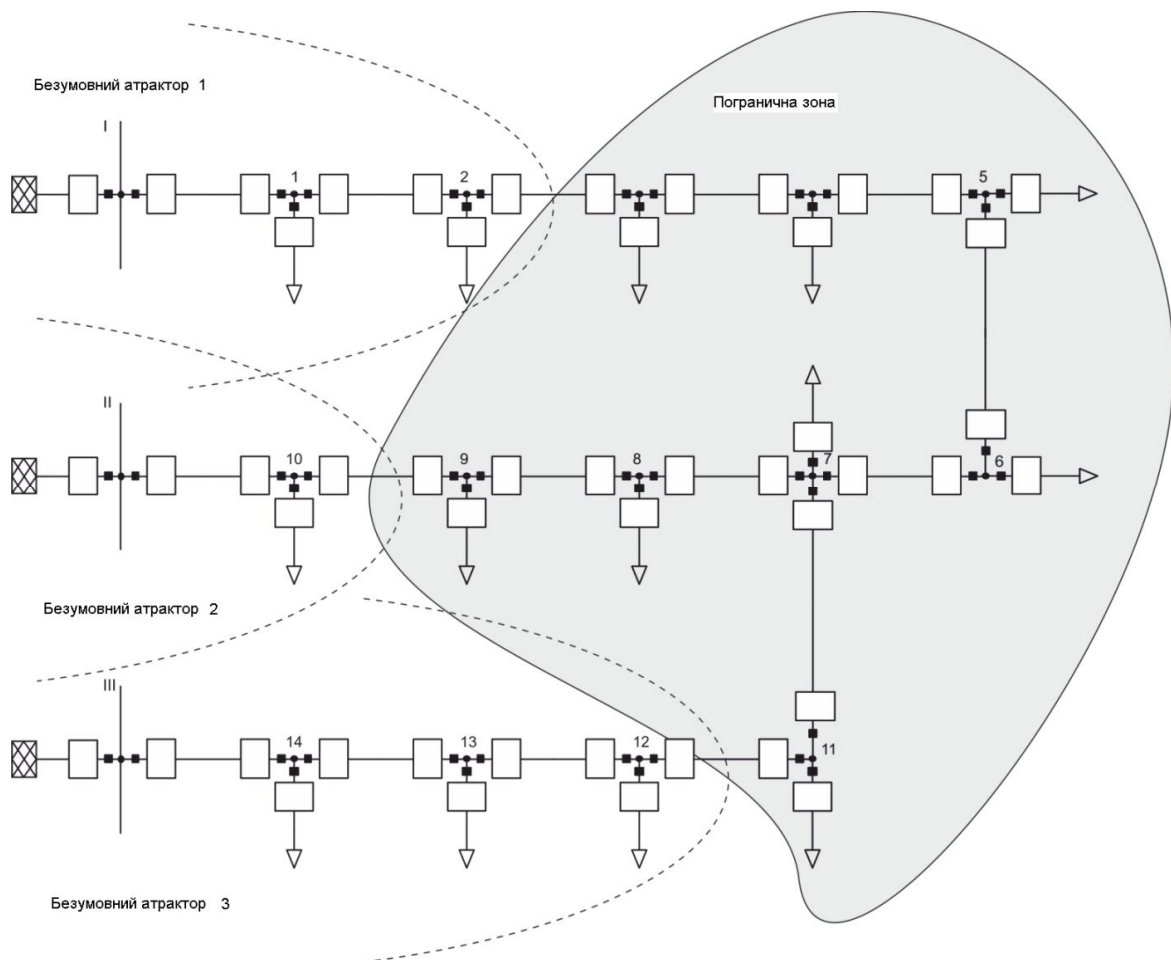


Рис. 8.9. Безумовні атрактори та погранична зона умовної електричної мережі з трьома лініями в програмному середовищі PowerFactory

2. Визначають спосіб кодування генетичної інформації про конфігурацію розподільної електричної мережі. Для цього формують бітовий ланцюжок, довжина якого дорівнює кількості ділянок у складі «пограничної» зони. Кожний біт такого ланцюжка визначає стан відповідної ділянки схеми («1» – ділянка знаходиться у роботі; «0» – ділянка є резервною перемичкою). Бітовий ланцюжок схеми пограничної зони розподільної електричної мережі з 52 біт у шістнадцятковій системі: **FFFFFDDFFFF7F**.

Нульові біти відповідають місцям розміщення розімкнених роз'єднувачів, що секціонують електричну мережу між трьома джерелами живлення.

3. Формують початкову популяцію можливих конфігураційних рішень (схем) розподільної електричної мережі. Обсяг початкової популяції залежить від розмірності «пограничної» зони схеми та перебуває у межах 20–30 особин. Формують фенотипи всіх конфігурацій схем, тобто особин початкової популяції. Одна з таких особин має відповідати вихідній схемі розподільної мережі. Решту фенотипів заповнюють за допомогою генератора випадкових чисел із рівномірним розподілом.
4. Для всіх особин початкової популяції конфігурацій можливих робочих схем обраховують значення прийнятої цільової (фітнес) функції, яка визначає ступінь оптимальності відповідного конфігураційного рішення. Також визначають усереднене значення цільової функції для початкової популяції в цілому.
5. Формують колесо «рулетки» для визначення поточної «батьківської» пари конфігурацій схем розподільної електричної мережі. Для цього умовне колесо поділяють на сектори. Кількість секторів дорівнює кількості особин поточної популяції. Ширина сектора має бути тим більшою, чим менше значення цільової функції для відповідного конфігураційного рішення. Ширину кожного сектора у відносних одиницях визначають за виразами

$$Z'_i = \frac{1}{Z_i}; \quad \Delta_i = \frac{Z'_i}{\sum_{i=1}^N Z'_i},$$

де Z_i – значення цільової функції для схеми поточної популяції; i – індекс поточної схеми (особини) у складі популяції; N – загальна кількість схем у складі популяції; Δ_i – ширина сектора «рулетки», який відповідає i -й схемі популяції у відносних одиницях.

6. За допомогою генератора випадкових чисел з рівномірним розподілом двічі «запускають» колесо «рулетки» та визначають дві особини поточної популяції, бітовий ланцюжок яких буде використано для формування нового дочірнього технічного рішення.

7. До обраної батьківської пари застосовують генетичний оператор схрещування (одно– або двоточкового кросинговеру), результатом виконання якого буде формування нового бітового ланцюжка (дочірнього рішення).
8. За допомогою генератора випадкових чисел із рівномірним розподілом визначають необхідність застосування генетичного оператора мутації до сформованого нового дочірнього рішення і у разі необхідності застосовують такий оператор.
9. Для нового конфігураційного рішення обчислюють значення цільової функції оптимальності.
10. Формують колесо «рулетки» для визначення особи поточної популяції, яку замінить генероване нове рішення. У цьому разі ширина секторів колеса має бути тим більшою, чим більше значення цільової функції відповідного фенотипу. Таку ширину у відносних одиницях обчислюють за виразом

$$\Delta_i = \frac{Z_i}{\sum_{i=1}^N Z_i}.$$

За допомогою генератора випадкових чисел із рівномірним розподілом виконують одноразовий «запуск» колеса «рулетки» та визначають особину поточної популяції, яку замінить нове дочірнє конфігураційне рішення. Виконують таке заміщення та перераховують усереднене значення цільової функції для оновленої популяції, тобто множини схем.

11. Перевіряють умови збіжності генетичного алгоритму. Якщо кількість поколінь генерованих дочірніх рішень досягла граничного значення або вичерпання часу, відпущеного на еволюцію, або всі бітові ланцюжки (фенотипи) поточної популяції можливих конфігурацій схем електричної мережі збіглися до одного вигляду (рішення), роботу генетичного алгоритму завершують, а найкраще (з найменшим значенням цільової функції) конфігураційне рішення схем поточної популяції приймають як оптимальне. В іншому разі виконують наступний цикл генетичного алгоритму, починаючи з процедури пункту 5.

8.3.1 Знаходження оптимальної конфігурації розподільної електричної мережі з використанням генетичного алгоритму

Розглянемо розв’язання задачі оптимальної реконфігурації розподільної електричної мережі на підставі розробленого методу з використанням апарата генетичних алгоритмів.

Описаний метод визначення оптимальної конфігурації схеми розподільної електричної мережі з використанням генетичного алгоритму застосуємо на підсистемі електричної мережі, яка утворена фідерами Л–4 і Л–18 ПС «Гнівень», Л–20 ПС «Гнівень» і Л–22 ПС «СЗБ» (рис.8.10).

Розімкнені роз'єднувачі Р-139, Р-163 та роз'єднувач на КТП 396 секціонують цю мережу на три частини, поділяючи споживачів між трьома лініями живлення. Нерівномірний розподіл навантаження між фідерами свідчить про неоптимальність конфігурації робочої схеми електричної мережі (табл. 8.2).

Зміну місць розміщення розімкнених роз'єднувачів необхідно виконувати шляхом зменшення сумарних втрат активної потужності в мережі.

Табл. 8.2. Режимні параметри мережі на номінальній напрузі 20 кВ

Фідер	Струм головної ділянки		Напруга на найвіддаленішій ТП	Втрати напруги	
	А	%	кВ	кВ	%
ПС «Гнівань»					
Л-2	4	2,1	21	0,001	0,005
Л-28	10	3,1	20,99	0,01	0,05
Л-4	70	25,7	20,81	0,19	0,95
Л-20	50	25,0	20,75	0,25	1,25
Л-22	10	3,9	20,99	0,01	0,05
ПС «СЗБ»					
Л-18	10	2,6	21	0,001	0,005
Л-12	3	1,5	21	0,001	0,005
Л-22	130	63,8	19,95	1,05	5,25
Л-20	30	11,8	20,92	0,08	0,4

На першому етапі застосування методу у складі розрахункової схеми слід виділити зони безумовних атракторів для кожного фідера та «пограничну» зону пошуку місця розміщення резервної перемички.

Для виділення безумовних атракторів джерел живлення розподільної електричної мережі застосуємо мінімаксий підхід. Для цього визначаємо електричні центри ділянок між можливими парами джерел живлення, збільшуючи навантаження до максимуму, з одного боку відповідної лінії з двобічним живленням та знижуючи до мінімуму навантаження з іншого боку (до прогнозованої перемички між фідерами). Будемо вважати, що в режимі мінімальних навантажень потужності споживачів дорівнюють 50 % своїх максимальних значень, що характерно для графіків навантажень типових споживачів сільських електричних мереж.

Виділення безумовних атракторів у підсистемі, яка утворена лінією з двобічним живленням фідерів ПС «Гнівань», показано на рис. 8.11. Тут S_e позначено еквівалентне навантаження споживачів на ділянках між ТП-202 та ТП-296, які отримують живлення від окремого відгалуження лінії з двобічним живленням.

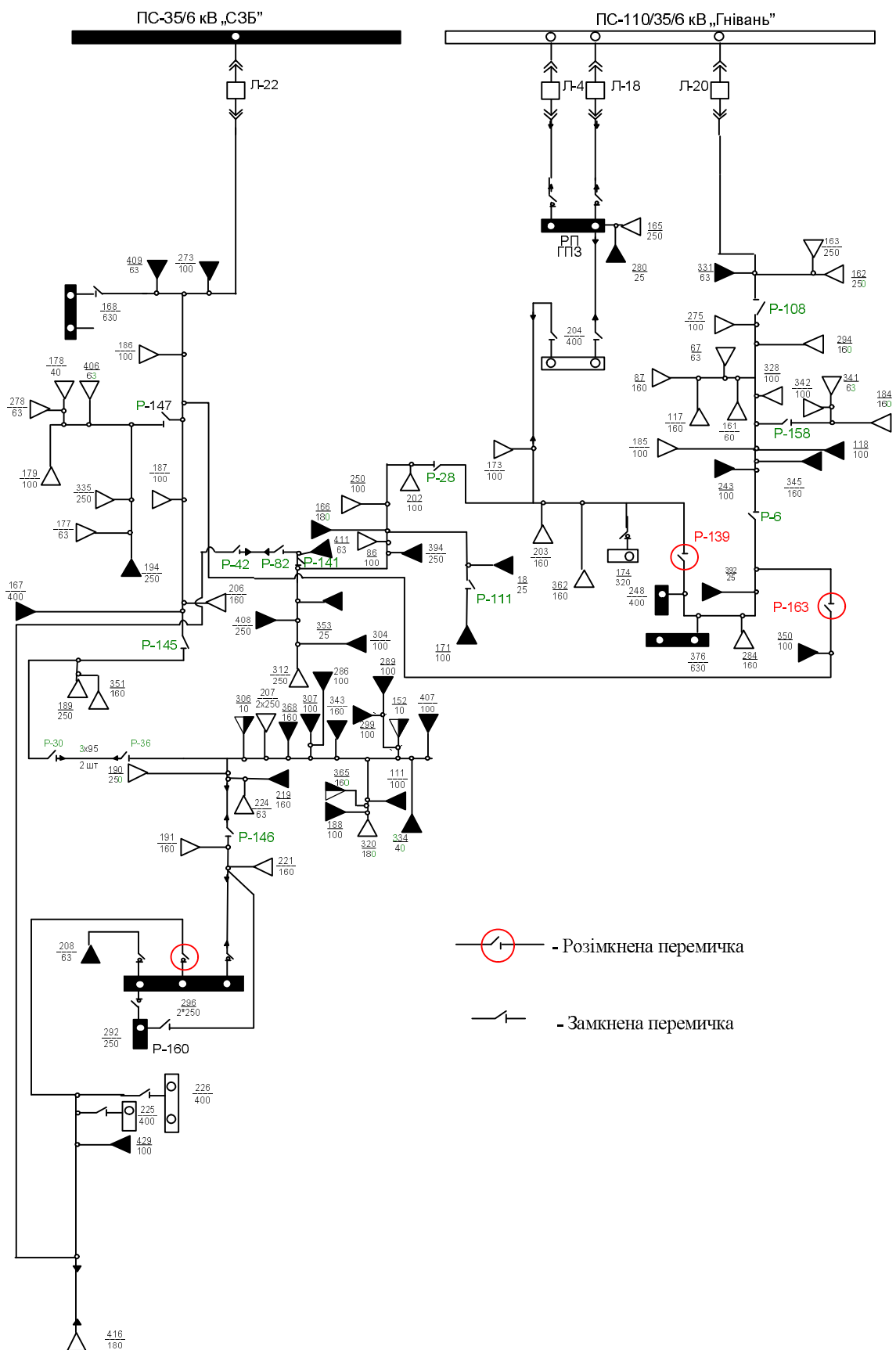


Рис. 8.10. Розподільна мережа напругою 6 кВ ПС «СЗБ» та ПС «Гнівань»

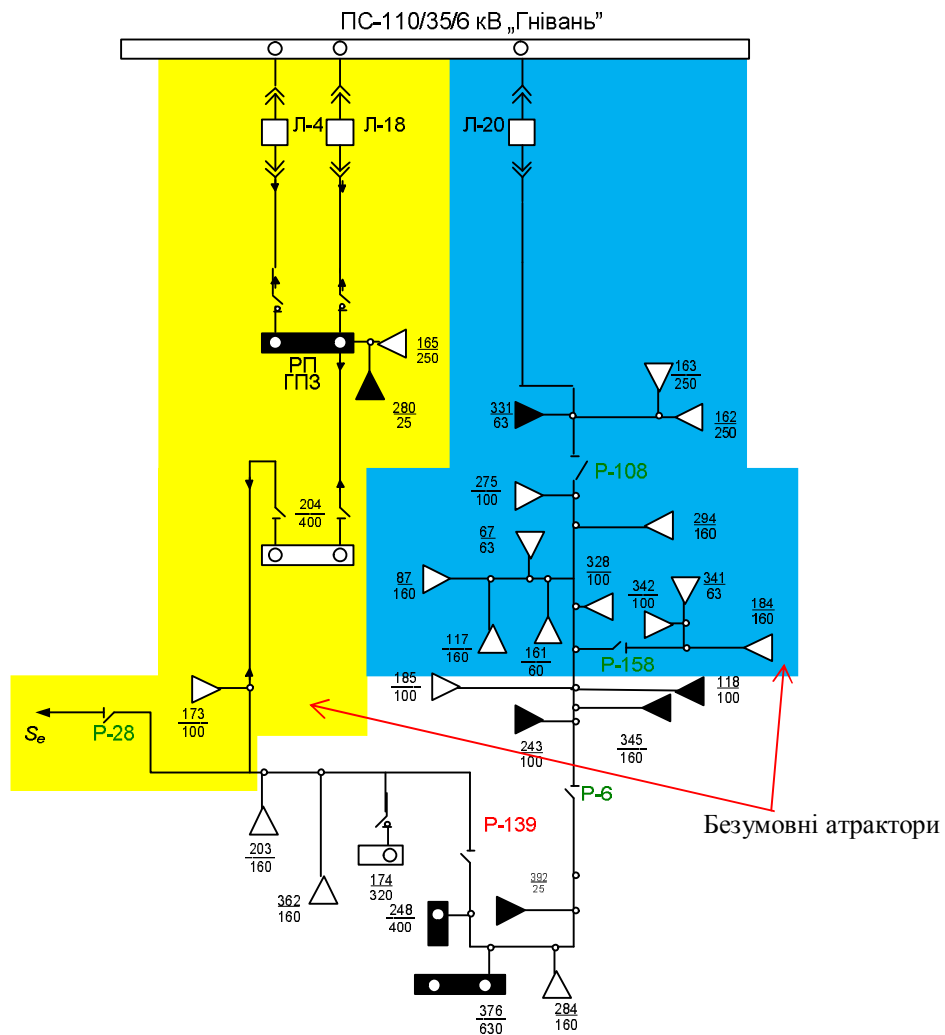


Рис. 8.11. Виділення безумовних атракторів фідерів ПС «Гнівань»

Для визначення безумовного атрактора фідерів Л-4+Л-18 необхідно збільшити до максимуму потужності навантажень найближчих до шин ПС «Гнівань» з боку даних фідерів споживачів, тобто навантаження ТП 165, 173, 174, 203, 204, 280, 362 та S_e . Водночас слід зменшити до мінімуму навантаження споживачів, найближчих до шин ПС «Гнівань» з боку фідера Л-20, тобто навантаження ТП 67, 87, 117, 118, 161, 162, 163, 184, 185, 243, 248, 275, 284, 294, 328, 331, 341, 342, 345, 376 та 392. Для такої розрахункової схеми електричний центр лінії з двобічним живленням належить ділянці між ТП-173 та ТП-203, визначаючи зону безумовного атрактора фідера Л-4+Л-18. Аналогічно визначаємо електричний центр лінії у разі, коли максимізовані потужності навантажень, найближчих до шин ПС «Гнівань» з боку фідера Л-20, одночасно мінімізуючи потужності навантажень, найближчих до шин ПС «Гнівань» з боку фідерів Л-4+Л-18. У цьому разі електричний центр належить ділянці між ТП-342 та ТП-185, визначаючи зону безумовного атрактора фідера Л-20.

Аналогічно визначають електричні центри інших ліній з двобічним живленням, утвореним можливими парами джерел живлення досліджуваної електричної мережі. Результати таких розрахунків зведено до табл. 8.3.

Табл. 8.3. Електричні центри фідерів за двобічного живлення

Фідер	Лінія з двобічним живленням	Ділянка, якій належить електричний центр навантажень
Л-4 + Л-18 (Гнівань)	Л-4+Л-18 (Гнівань) – Л-20 (Гнівань)	ТП-173– ТП-203
	Л-4+Л-18 (Гнівань) – Л-22 (СЗБ)	ТП-225 – ТП-226
Л-20 (Гнівань)	Л-20 (Гнівань) – Л-4+Л-18 (Гнівань)	ТП-342 – ТП-185
	Л-20 (Гнівань) – Л-22 (СЗБ)	Роз'єднувач Р-6
Л-22 (СЗБ)	Л-22 (СЗБ)– Л-4+Л-18 (Гнівань)	Роз'єднувач Р-145
	Л-22 (СЗБ)– Л-20 (Гнівань)	ТП-186 – ТП-150

Поєднання результатів визначення електричних центрів всіх можливих ліній з двобічним живленням обумовлює зони безумовних атракторів всіх джерел живлення та «пограничної» зони, наведених на рис. 8.12.

На рис. 8.13 показано «пограничну» зону атракторів, в якій буде здійснюватись пошук місць розміщення розімкнених резервних перемичок, які будуть оптимально секціонувати розподільну мережу з метою мінімізувати втрати активної потужності на передавання електричної енергії до споживачів. «Погранична» зона містить 52 ділянки ліній електропередавання повітряного та кабельного виконання, які забезпечують живлення 38 ТП. На рис. 8.13 показано положення розімкнених резервних перемичок, які відповідають вихідній схемі розподільної мережі та обумовлюють сумарні навантажувальні втрати активної потужності в мережі 190 кВт.

Мета оптимізації конфігурації робочої схеми розподільної мережі полягає у визначенні такого положення резервних перемичок, щоб втрати активної потужності на передавання електричної енергії були мінімальними за умови забезпечення обмежень щодо мінімальної зв'язності робочої схеми, забезпечення нормованої якості електричної енергії в точках приєднання абонентів відповідно до цільової функції.

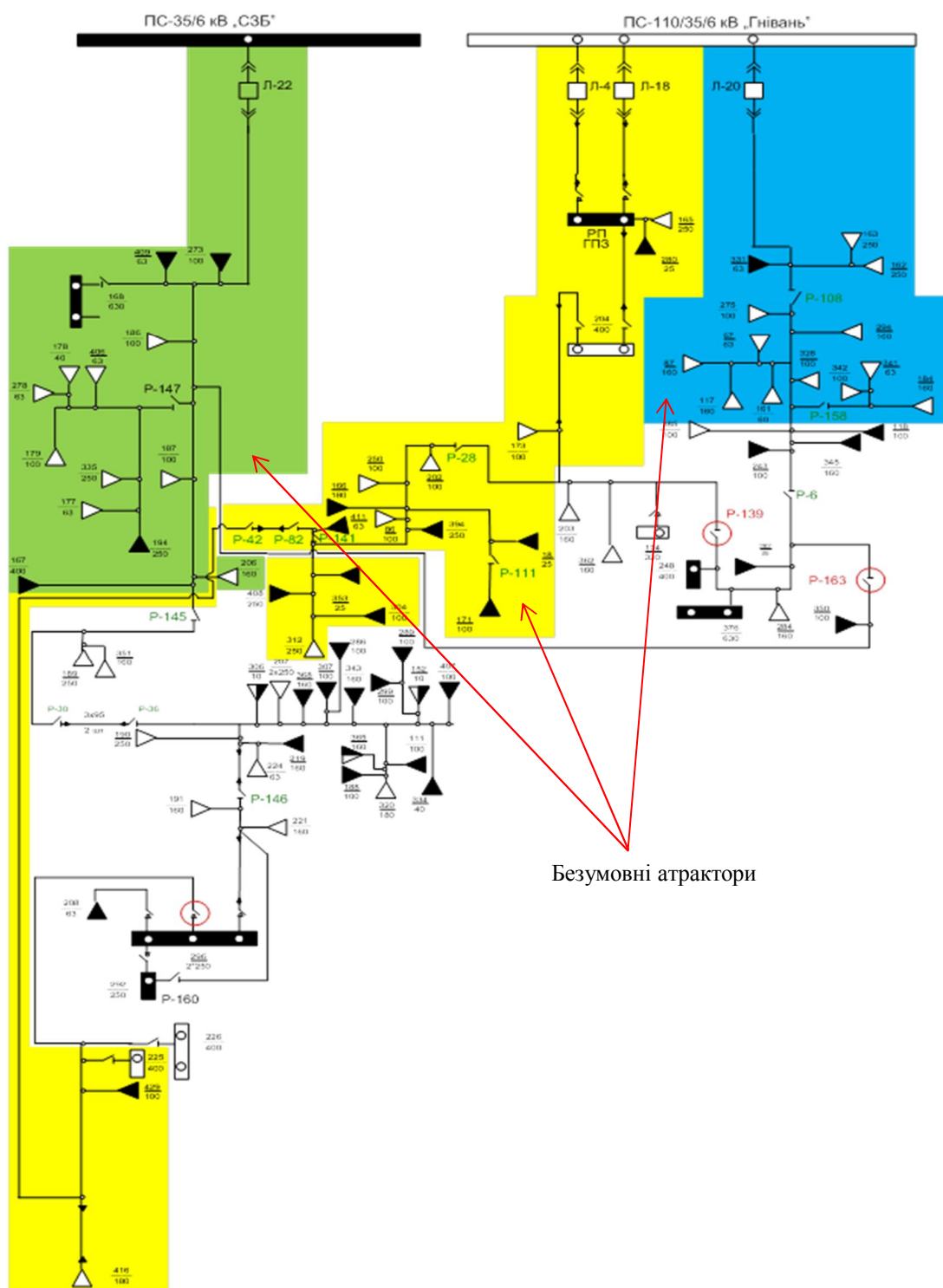


Рис. 8.12. Виділення безумовних атракторів трьох фідерів та «пограничної» зони розподільної електричної мережі

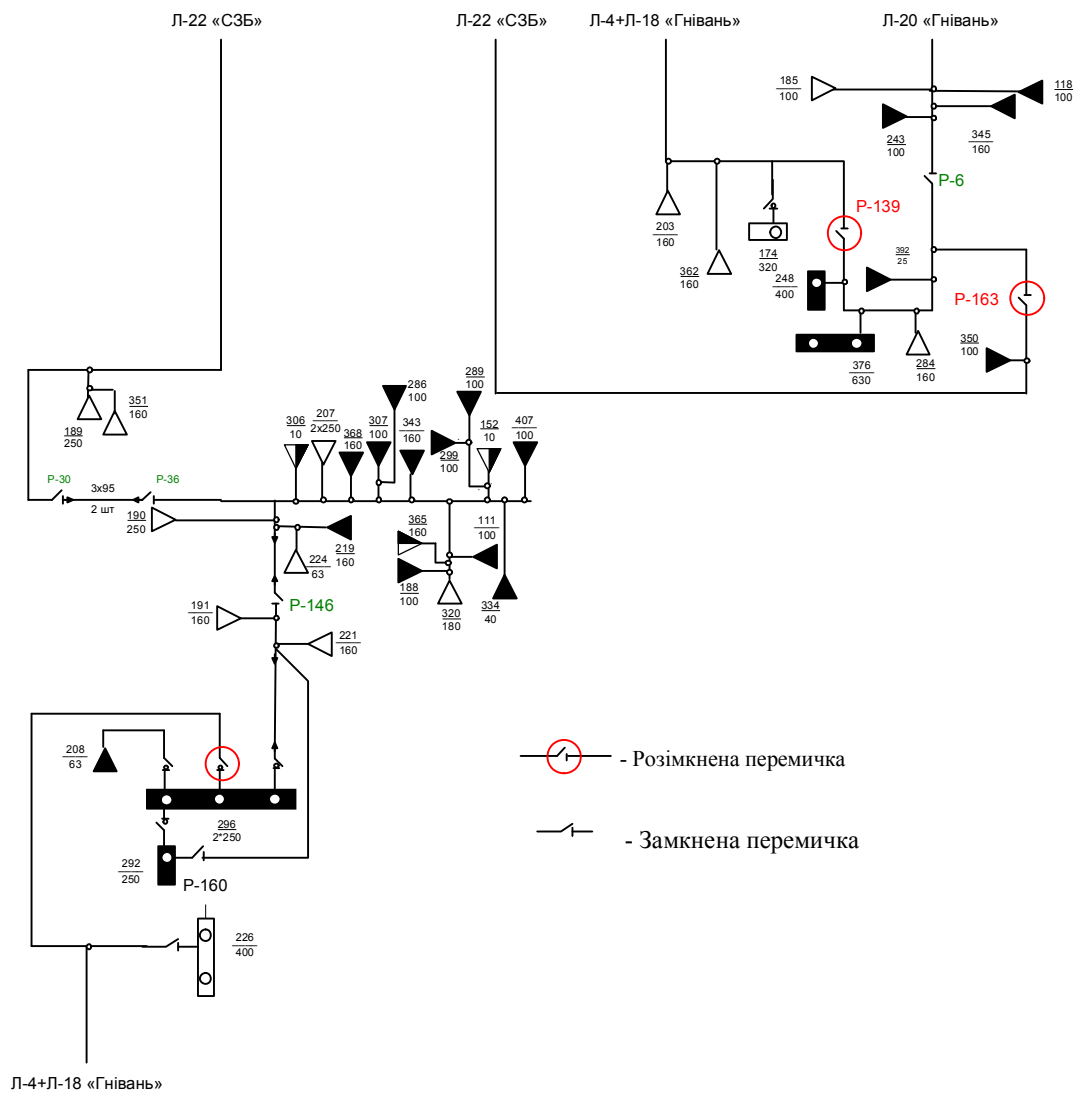


Рис. 8.13. «Погранична» зона розподільної електричної мережі за поточного положення розімкнених резервних перемичок

Реконфігурація робочої схеми розподільної електричної мережі полягає у переміщенні положення розімкненої перемички. Формально стан кожної ділянки робочої схеми можна представити за допомогою двійкового коду:

- «0» – якщо ділянка є розімкненою перемичкою та не входить до складу тракту покриття навантажень під'єднаних абонентів;
- «1» – якщо ділянка знаходиться в роботі.

Так, конфігурацію «пограничної» зони атракторів джерел живлення досліджуваної електричної мережі, яка містить 52 ділянки, можна подати за допомогою 52-бітного ланцюжка, кожний біт якого містить інформацію про стан відповідної ділянки схеми. Отже, загальний простір пошуку складається із 2^{52} можливих станів. Для наведення генетичної інформації конфігурації схем взято 52-бітну структуру (13 ген по чотири біти), що дає змогу використовувати шістнадцяткову систему для кодування конфігурації можливої схеми електричної мережі. Так, для наведення генетичної інформації можливих

конфігураційних рішень можна використовувати 13-символьний набір, кожний символ якої в шістнадцятирічній системі матиме 4-бітну генетичну інформацію. Так, наприклад, схема досліджуваної мережі за поточного положення розімкнених резервних перемичок закодована таким бітовим рядком:

«FFFEFFDFFFF7F».

Червоним кольором позначено символи, коди яких містять нульові біти, яким відповідають місця розміщення розімкнених роз'єднувачів, що секціонують електричну мережу між трьома джерелами живлення.

Для оцінювання ефективності породжуваних технічних рішень як цільову функцію застосовано функцію оптимальності (8.1). Очевидно, що наведена форма цільової функції відповідає мінімізаційній задачі, тобто кращим технічним рішенням відповідають менші числові значення цільової функції. Цю обставину слід врахувати під час формування батьківських пар методом рулетки та під час визначення особини, яку замістить породжуване дочірнє рішення.

Початкова популяція генетичного алгоритму сформована з 20 особин, одна з яких відповідає конфігурації робочої схеми за поточного положення розімкнених резервних перемичок, генотипи всіх інших заповнено за допомогою генератора випадкових чисел. Виконання генетичного алгоритму проілюстровано на діаграмі рис. 8.14. Тут крива $Z_{\min}(g)$ є динамікою зміни значення оціночної функції оптимальності в поколіннях для найкращого технічного рішення; $Z_{\text{сер}}(g)$ – динаміка зміни усередненого значення оціночної функції для всього покоління в цілому.

З аналізу кривої графіка (рис. 8.14) видно, що після виконання 30 повних циклів генетичного алгоритму, значення цільової функції для найкращої особини перестало змінюватися, що свідчить про досягнення збіжності процесу пошуку. Оптимальна конфігурація досліджуваної електричної мережі наведена на схемі рис. 8.14. Така схема характеризується сумарними втратами активної потужності 0,17 МВт.

Таким чином, з метою вирівнювання навантаження між окремими фідерами та зниження втрат активної потужності на основі генетичного алгоритму сформована конфігурація нормальної схеми мережі, яка полягає в такому:

- 1) живлення КТП № 248, 376, 392 перенести з фідера Л–20 ПС «Гнівань» на фідер Л–4 ПС «Гнівань». Для цього перемичку Р–139 перенести з ділянки між КТП–174 та КТП–245 на ділянку між КТП–376 та КТП–284;
- 2) живлення КТП № 175, 284 перенести з фідера Л–20 ПС «Гнівань» на фідер Л–22 «СЗБ». Для цього роз'єднувач Р–163 в нормальному режимі замкнути, а роз'єднувач Р–6 розімкнути;

- 3) живлення КТП № 208, 296, 292, 221, 191, 219, 224, 190, 306, 207, 368, 307, 286, 152, 299, 289, 343, 320, 188, 189, 365, 334, 351, 407 перенести з фідера Л-22 ПС «СЗБ» на фідер Л-4 ПС «Гнівань». Для цього роз'єднувач ВН ТП-296 у нормальному режимі замкнути та розімкнути роз'єднувачі Р-160 та Р-36.

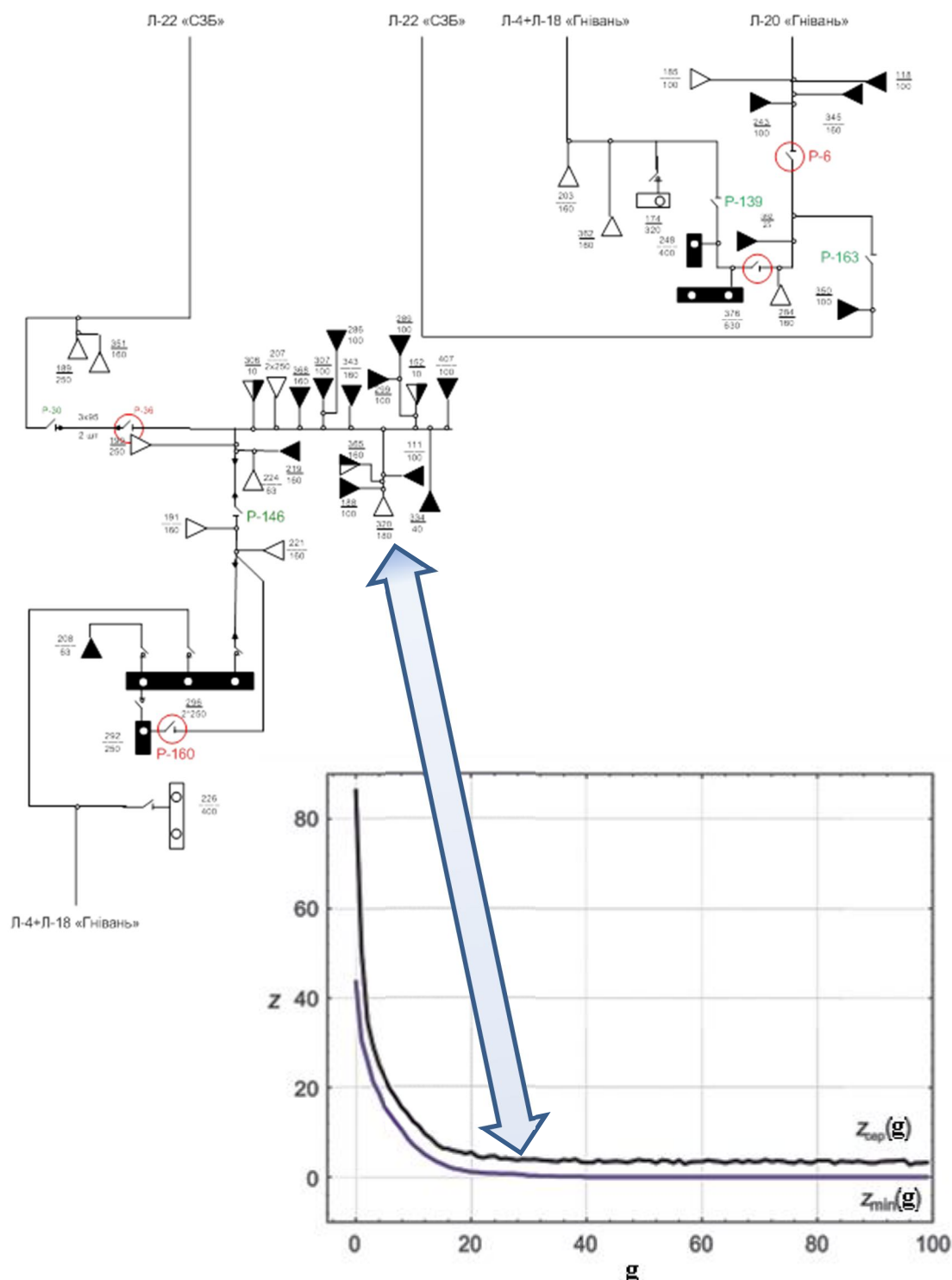


Рис. 8.14. Фрагмент оптимальної схеми електричної мережі в тридцятому поколінні з кривими зміни значення функції оптимальності в поколіннях $Z_{\min}(g)$, $Z_{\text{сер}}(g)$

Проведені експериментальні розрахунки з оптимізації конфігурації розподільної електричної мереж підтвердили ефективність запропонованого методу оптимальної реконфігурації розподільної електричної мережі з використанням генетичних алгоритмів. Зокрема, оптимальна реконфігурація розрахункової розподільної мережі дала змогу знизити сумарні втрати активної потужності на 10 % (з 0,19 МВт до 0,17 МВт), у разі забезпечення належної якості та надійності електропостачання. Під час реалізації алгоритму розглянуто до 600 варіантів конфігурацій, що становить менше 0,01% загального простору пошуку. Розрахунковий час знаходження оптимальної конфігурації склав: для схеми зі 188 ділянок – 90 секунд; для 88 ділянок – 5 секунд.

8.4 Навчальна програмна реалізація генетичного алгоритму

Для реалізації математичної моделі генетичного алгоритму існує декілька програм, наприклад, GeneHunter, Genetic Training Option, Evolver, Easy NP, Egalt та інші.

Розглянемо структурну, а також параметричну ідентифікацію виконання генетичного алгоритму (ГА) на прикладі програми EGALT.

Програма має дружній графічний інтерфейс користувача і дає змогу студентам реалізувати ГА та керувати його виконанням.

Програмний пакет EGALT написаний мовою C++ та розвинутий на платформі Windows NT. На цьому етапі – це програма платформи TM Win32. Процес реалізації генетичного алгоритму забезпечує тільки один файл «EGALT.EXE». Він може бути запущеним «подвійним натисканням» на файлі EGALT.EXE у диспетчері файлів Windows. Основне вікно з'явиться як графічний інтерфейс користувача EGALT. На рис. 8.15 показано початковий вигляд вікна EGALT. Всі відповідні параметри, наведені на передній панелі вікна, інтуїтивно зрозумілі користувачу і дають зворотний зв'язок у реальному часі з процесом прийняття рішення.

Верхня зона передньої панелі представляє інформацію під час виконання ГА. Шість складових зони наведені нижче:

Generation: номер покоління поточної популяції;

Timer: час роботи ГА;

Speed: число генерованих поколінь за секунду;

Best: мінімальне значення функції, що знайдено на поточний час;

Average: середнє значення цільової функції;

Object: початкове значення або значення цільової функції, досягнуте на поточний час;

Best chromosome: ген кращого (отриманого) рішення. Картографія між генним значенням та його кольором показана в карті Gene map. Користувачі можуть «натиснути» на ген, щоб побачити точне значення цього гена;

Gene map: відтворює картографію між генним значенням та його кольором.

Вибір функцій (Function)

На рис. 8.16 показано вибір цільових функцій та доступних опцій у вікні програми EGALT.

EGALT містить одинадцять цільових функцій оптимізації, які мають бути мінімізовані. Ці функції покривають широкий діапазон галузей застосування, у тому числі й електроенергетику.

Цільова функція – це унарна функція. Формули доступних функцій F0 – F10 наведені нижче:

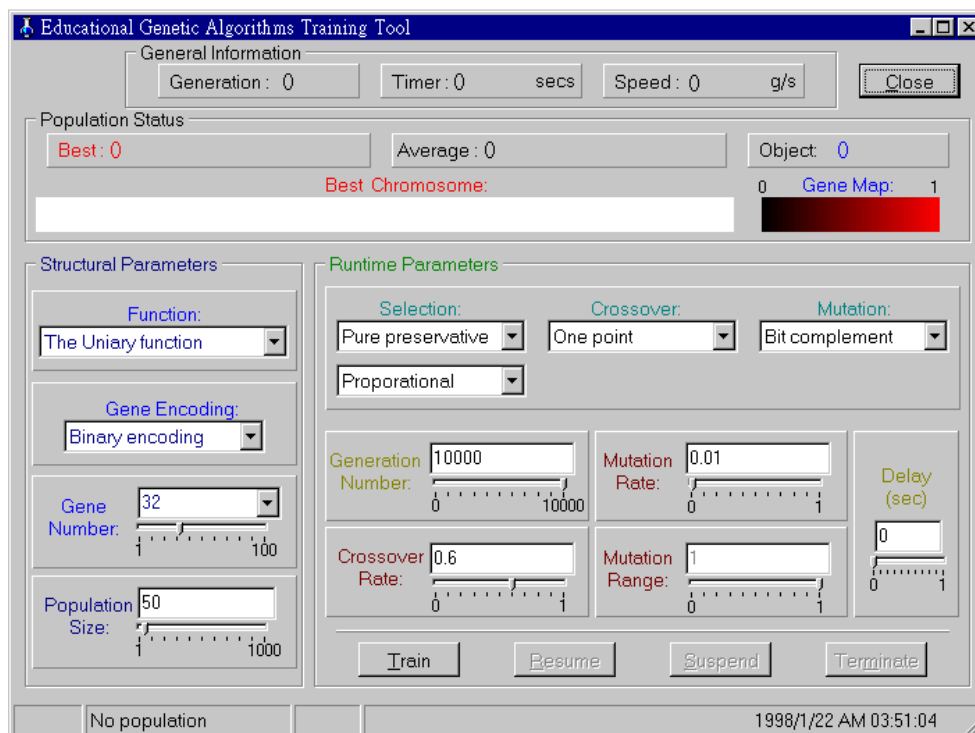


Рис. 8.15. Вікно ініціалізації програми EGALT

F0: $f_0(\bar{x})$ = кількість нулів в $\bar{x}$ з $x_i \in \{0,1\}$, значення інтервалу: 0

Наприклад, якщо $\bar{x}_1$ – це чотиризначний вектор з елементами 0, 1, 0, 0, то $f_0(\bar{x}_1)$ має бути 1. Оптимальний рівень $f_0(\bar{x})$ – це 0 для всіх просторових (розмірних) версій $\bar{x}$.

Друга, третя, четверта, п'ята та шоста функції запозичені з «First International Contest on Evolutionary Optimization» («Перший Міжнародний Конкурс з Еволюційної Оптимізації»). Перший конкурс еволюційної оптимізації було

проведено 1996 IEEE International conference on Evolutionary Computation (ICEC'96), May 20–22, 1996, Nagoya, Japan.

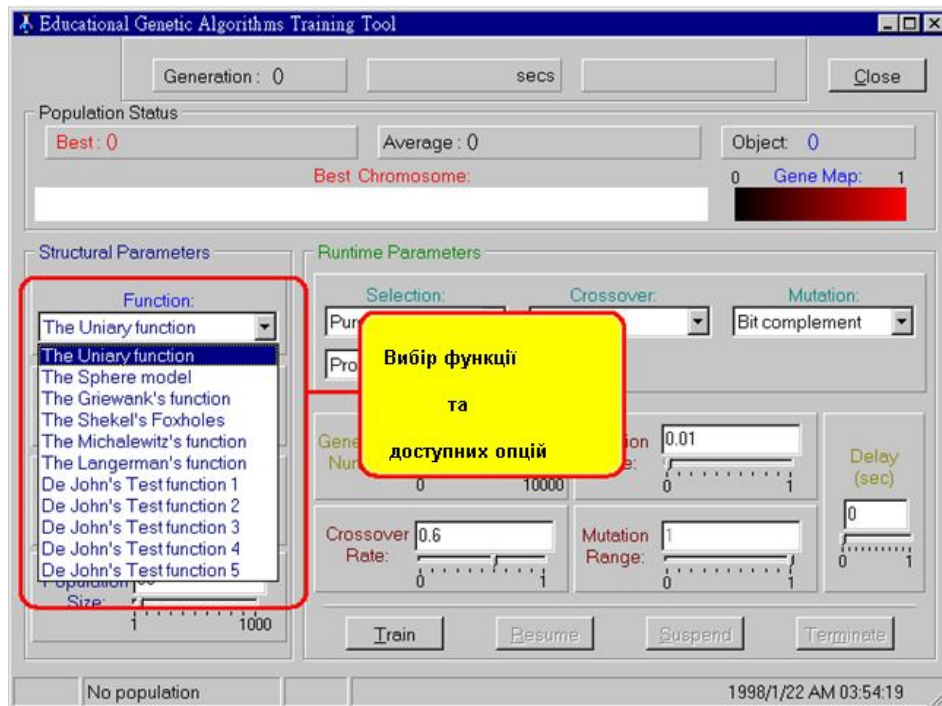


Рис. 8.16. Перевірка вибору функцій та доступних опцій

F1: "The sphere model", формула якої має такий вигляд:

$$f_1(\bar{x}) = \sum_{i=1}^N (x_i - 1)^2,$$

з $x_i \in [-5, 5]$, значення досягає: $1.0\text{E-}6$.

F2: "Griewank's function", формула якої має вигляд:

$$f_2(\bar{x}) = \frac{1}{4000} \sum_{i=1}^N (x_i - 100)^2 - \prod_{i=1}^N \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1,$$

з $x_i \in [-600, 600]$, значення досягає: $1.0\text{E-}4$.

F3: "Shekel's foxholes", формула якої така:

$$f_3(\bar{x}) = - \sum_{i=1}^{30} \frac{1}{\|\bar{x} - A(i)\|^2 + c_i},$$

з $x_i \in [0, 10]$, значення досягає: -9 .

F4: "Michalewicz's function", формула якої така:

$$f_4(\bar{x}) = - \sum \sin(x_i) \times \sin^{20},$$

з $x_i \in [0, \pi]$, значення досягає: -4.687 для п'ятизначних версій, -9.66 для десятизначних версій.

F5: "Langerman's function", формула якої така:

$$f_5(\bar{x}) = - \sum c_i \left(e^{-\frac{1}{\pi} |x - A(i)|^2} \times \cos(\pi \times \|\bar{x} - A(i)\|^2) \right),$$

з $x_i \in [0, 10]$, значення досягає: -1.4

F6: $f_6(\bar{x}) = \sum_{i=1}^N x_i^2$, з $x_i \in [-5.12, 5.12]$, значення досягає: 1E-4.

Сьома, восьма, дев'ята, десята та одинадцята перевірючі функції запозичені з De Jong's аналізу, "An Analysis of the Behavior of a Class of Genetic Adaptive Systems" ("Аналіз Поведінки Класу Генетичних Адаптивних Систем")⁴:

F7: $f_7(\bar{x}) = 100 \times (x_1^2 - x_2)^2 + (1 - x_1)^2$, з $x_i \in [-2.048, 2.048]$,

значення досягає 1E-4.

F8: $f_8(\bar{x}) = \sum_{i=1}^N |x_i|$, з $x_i \in [-5.12, 5.12]$, значення досягає: 0.

F9: $f_9(\bar{x}) = \sum_{i=1}^N (i \times x_i)$, з $x_i \in [-1.28, 1.28]$, значення досягає: 1E-4.

F10:

$$f_{10}(\bar{x}) = \frac{1}{0.002 + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a_{ij})^6}},$$

з $x_i \in [-65.536, 65.536]$, значення досягає: 9980.

⁴ K. A. De Jong. *An Analysis of the Behavior of a Class of Genetic Adaptive Systems*. Ph.D. dissertation, University of Michigan, 1975.

Вибір методу кодування (Encoding method)

Методи кодування в EGALT такі: бінарне кодування, Gray кодування і Naïve кодування. Бінарне кодування є подвійним типом мовою програмування C. Код є 64-розрядним рядком на платформі Win32 TM. На рис. 8.17 показано вікно вибору кодування. Бінарне кодування – це початковий метод кодування.

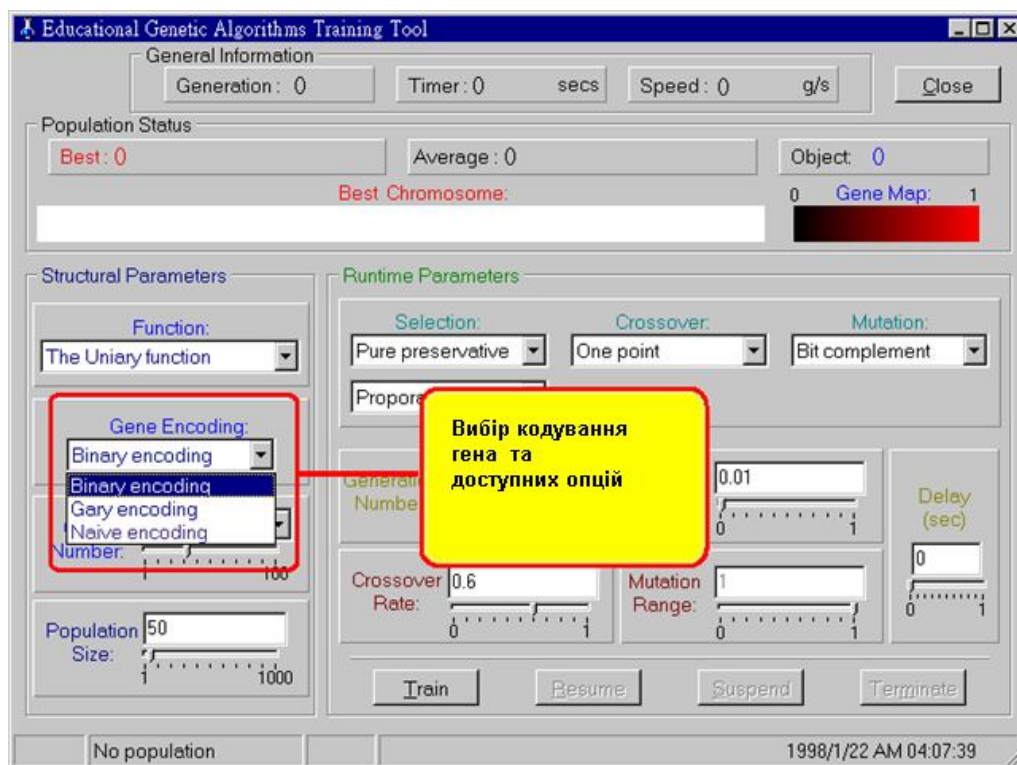


Рис 8.17. Селектор представлення і доступні опції

Вибір кількості генів (Gene number)

«Gene number» є кількістю генів у хромосомі. У більшості випадків більше число генів – це важча проблема вирішення для ГА. Оскільки кількість генів є розмірністю виміру в проблемі, то великий вимір є значною зоною пошуку. *Gene number* приймається в діапазоні з 1 до 100 і можливо фіксується в заданих значеннях для деяких перевіряючих функцій. Наприклад, якщо перевіряюча функція, яку ви вибрали, є De Jong's testing function 2, F7, в ній кількість генів буде постійно дорівнювати 2. Вибір починаємо з функціональних потреб тільки двох елементів x_1 та x_2 . Початкова кількість генів залежить від цільової функції.

Вибір розміру популяції (Population size)

«Population size» є кількістю індивідумів у поколінні. Розмір популяції приймається з 1 до 1000. Розмір популяції впливає як на результативність, так і ефективність ГА. Генетичні алгоритми з малими популяціями загалом виконуються погано, оскільки популяція передбачає недостатній об'єм вибірки для найбільших гіперплощин. Вірогідніше, велика популяція має представників від великого ряду гіперплощин. До того ж велика популяція може запобігти передчасному сходженню генетичних алгоритмів, але це вимагає більше оцінок за покоління. Початковий розмір популяції береться до 50.

Вибір опції схрещування або селекції (Selection)

Опції кожного оператора вибору діляться на дві частини: основна і другорядна. В основній частині є три доступні опції: пропорційний відбір (proportional selection), класифікаційний вибір (ranking selection) і двійковий турнірний вибір (binary tournament selection). Другорядна частина передбачає додаткові характеристики вибору. У EGALT ми передбачаємо чистий консервант (pure preservative), елітарний консервант (elitist preservative), правого вимирання (right extinctive) і лівого вимирання (left extinctive). Повний оператор вибору має містити як другорядні, так і головні опції. Наприклад, чистий запобіжний пропорційний оператор вибору комбінує опції «чистий консервант» і опції «члена» пропорції. Три головних опції вибору, об'єднані з чотирма додатковими, дають можливість реалізації до дванадцяти видів операторів вибору. Чистий консервант пропорційної опції – це початковий оператор вибору.

Вибір кросовера або кросинговера (Crossover)

Програма має чотири оператори кросовера: одноточковий (one-point crossover), двоточковий (two-point crossover), багатоточковий (multi-point crossover) і однорідний (uniform crossover). У разі моделювання можна заборонити кросовер оператор, приводячи швидкість до нуля (None). One-point crossover – заданий за замовчуванням.

Вибір мутації (Mutation)

Оператор мутації буде встановлений автоматично згідно з кодуванням гена. Якщо гени кодуються в бінарному кодуванні або Gray кодуванні, то оператор мутації буде встановлений до «розрядного доповнення» (bit

complement). З іншого боку, якщо використовується Naïve кодування, то оператор мутації буде шумовим – змішаним (Noise). Оператор мутації можна заборонити, приводячи частоту мутацій до нуля (None).

Коефіцієнт кросовера (Crossover rate)

Коефіцієнт кросовера задає частоту, з якою використовується кросовер оператор. У кожному поколінні $Cr \cdot N$ представляє індивідуми кросовера. Можливо встановити коефіцієнт кросовера в діапазоні між 0 і 1. Заданий за замовчуванням коефіцієнт кросовера складає 0,6.

Коефіцієнт мутації (Mutation rate)

У разі використання методів кодування (бінарне кодування і Gray кодування) бітовий рядок має приблизно $Mr \cdot N \cdot L$ кількість мутацій, які відбуваються за покоління. А за використання Naïve методу кодування маємо приблизно $Mr \cdot N$ мутацій за покоління. Можна встановити коефіцієнт мутації в діапазоні між 0 і 1. За замовчуванням коефіцієнт мутації складає 0,01.

Діапазон мутації (Mutation range)

Діапазон мутації використовується тільки за Naïve кодування. Діапазон мутації має ширину білого шумового діапазону для генів, які вибрані для видозміни. Якщо діапазон великий, то популяція отримає більшу необ'єктивність після мутації.

Затримка (Delay)

Це опція використовується для управління швидкістю алгоритма. У менш складних випадках програма, можливо, виконується занадто швидко для оператора, щоб спостерігати поведінку ГА. Тому можливо додати деякої затримки в кожному поколінні.

КОНТРОЛЬНІ ЗАПИТАННЯ ТА ЗАВДАННЯ

1. Переваги генетичних алгоритмів перед класичними оптимізаційними методами.
2. Дайте визначення та характеристику генетичного алгоритму.
3. Дайте визначення поняттю хромосома в генетичному алгоритмі.
4. Застосування генетичних алгоритмів в електроенергетиці.
5. Дайте характеристику етапів генетичного алгоритму.
6. Вибір початкової популяції.

7. Кросинговер, схрещування та мутації в генетичному алгоритмі.
8. Дайте визначення та характеристику генетичного алгоритму. Дайте характеристику етапів генетичного алгоритму.
9. Вибір початкової популяції.
10. Кросинговер, схрещування та мутації в генетичному алгоритмі.
11. Застосування генетичних алгоритмів в електроенергетиці

ЛІТЕРАТУРА

[15], стор. 11–28; [16], стор. 6–20; [24], стор. 99–130

СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ

1. Tanaka K. Fuzzy Control Systems Design and Analysis: A Linear Matrix Inequality Approach / K. Tanaka, H.O.Wang. – New York: John Wiley & Sons, Inc., 2001. – 305 p.
2. Von Altrock C. Fuzzy Logic and NeuroFuzzy Applications Explained / Constantin von Altrock. – Upper Saddle River, New Jersey: Prentice Hall PTR, 1995. – 350 p.
3. Алиев Р. А. Интеллектуальные роботы с нечеткими базами знаний / Р. А. Алиев. – М. : Радио и связь, 1995. – 176 с.
4. Баранов Г. Л. Структурное моделирование сложных динамических систем / Г. Л. Баранов, А. В. Макаров. – Киев: Наук. думка, 1986. – 272 с.
5. Гостев В. И. Нечеткие регуляторы в системах автоматического управления / В. И. Гостев. – Киев: Радіоаматор, 2008. – 972 с.
6. Дьяконов В. Математические пакеты расширения MATLAB: специальный справ. / В. Дьяконов, В. Круглов. – СПб. : Питер, 2001. – 480 с.
7. Жданов А. А. Автономный искусственный интеллект / А. А. Жданов. – 3-е изд. – М.: БИНОМ. Лаборатория знаний, 2012. – 359 с.
8. Заде Л. Понятие лингвистической переменной и его применение к принятию приближенных решений / Л. Заде. – М. : МИР, 1976. – 165 с.
9. Кирик В. В. Інтелектуальні технології управління [Текст]: навч. посіб. / В. В. Кирик. – Київ, 2010. – 88 с.
10. Кирик В. В. Комп'ютерні технології моделювання та дослідження електротехнічних систем [Текст]: Монографія / В. В. Кирик, В. А. Халіков. – Київ, 2009. – 213с.
11. Колесников А. А. Синергетические методы управления сложными системами: Теория системного синтеза / А. А. Колесников. – М. : Ком–Книга, 2006. – 228 с.
12. Леоненков А. В. Нечеткое моделирование в среде MATLAB и fuzzyTECH / А. В. Леоненков. – СПб. : БХВ–Петербург, 2003. – 736 с.
13. Никифоров И. К. Альтернативные принципы управления в электроэнергетике на основе синергетического подхода / И. К. Никифоров // Электротехника. – 2013. – №12. – С. 15–18.
14. Пегат А. Нечеткое моделирование и управление [пер. с англ.] / А. Пегат. – М. : БИНОМ, Лаборатория знаний, 2009. – 798 с.
15. Ротштейн О. П. Інтелектуальні технології ідентифікації: нечіткі множини, генетичні алгоритми, нейронні мережі / О. П. Ротштейн. – Вінниця: УНІВЕРСУМ–Вінниця, 1999. – 320 с.

16. Рутковская Д. Нейронные сети, генетические алгоритмы и нечеткие системы / Д. Рутковская, М. Пилиньский, Л. Рутковский. – М. : Горячая линия – Телеком, 2007. – 452 с.
17. Саати Т. Аналитическое планирование. Организация систем [пер. с англ.] / Т. Саати, К. Кернс. – М. : Радио и связь, 1991. – 224 с.
18. Бабаков Н. А. Теория автоматического управления. Часть первая. Теория линейных систем автоматического управления [Текст]: Учеб. изд. / Н. А. Бабаков, А. А. Воронов, А. А. Воронова. – М. : Высшая школа, 1986. – 368 с.
19. Хайкин С. Нейронные сети: полный курс. 2-е изд. [пер. с англ.] / Саймон Хайкин. – М. : ООО «И.Д. Вильямсе», 2016. – 1104 с.
20. Штовба С. Д. Проектирование нечетких систем средствами MATLAB / С. Д. Штовба. – М. : Горячая линия – Телеком, 2007. – 288 с.
21. Электрические системы: Управление переходными режимами электроэнергетических систем [Текст]: Учеб. изд. / Под ред. В.А. Веникова, Э.Н. Зуева и М.Г. Портного. – Москва: Высшая школа, 1982. – 447 с.
22. Щерба А. А. Системи з нечіткою логікою регулювання електроенергетичних режимів / А. А. Щерба, В. В. Кирик. – К. : Лазурит–Поліграф, 2011. – 329 с.
23. Мандель Н. Д. Кластерный анализ / Н. Д. Мандель. – М. : Финансы и статистика, 1988. – 176 с.
24. Кирик В. В. Розподільні електричні мережі напругою 20 кВ та ефективність їх роботи / В. В. Кирик, Б. В. Циганенко, О. С. Яндульський. – Київ: КПІ ім. Ігоря Сікорського, Вид-во «Політехніка», 2018. – 228 с.
25. GeneHunter [Електронний ресурс] – Режим доступу: <http://www.wardsystems.com/manuals/genehunter/>. – Назва з екрана.
26. NeuroShell2 [Електронний ресурс] – Режим доступу: <http://www.wardsystems.com/manuals/neuroshell2/>. – Назва з екрана.
27. Основные положения концепции интеллектуальной энергосистемы с активно-адаптивной сетью [Электронный ресурс] – Режим доступа: <http://www.optima.ru/>.
28. Смолин Д. В. Введение в искусственный интеллект: конспект лекций / Д. В. Смолин. – М.: Физматлит, 2004. – 208 с.
29. Зайченко Ю. П. Основи проектування інтелектуальних систем: навч. посіб. / Ю. П. Зайченко. – К. : Вид. дім «Слово», 2004. – 352 с.
30. Аналіз зарубіжної практики впровадження автоматизованих систем управління технологічними процесами в електроенергетиці [Електронний ресурс] – Режим доступу: [//ua.energy/wp-content/uploads/2018/01/2.-SMART-GRID.pdf](http://ua.energy/wp-content/uploads/2018/01/2.-SMART-GRID.pdf).

31. Искусственный интеллект: справочник / Под ред. Д. А. Поспелова. – М. : Радио и связь, 1990. – 304 с.
32. Таран Т. А. Штучний інтелект. Теорія і застосування: навч. посіб. / Т. А. Таран, Д. А. Зубов. – Луганськ: Вид-во СНУ ім. В. Даля, 2006. – 242 с.
33. Інтелектуальні системи підтримки прийняття рішень [Електронний ресурс] – Режим доступу: [//studfiles.net/preview/5195121/page:3/](http://studfiles.net/preview/5195121/page:3/).
34. IEC 1131-7 Programmable Controllers. Fuzzy Control Programming/ Committee Draft CD 1.0. – Rel. 19 Jan 97 [Електронний ресурс] – Режим доступу: <http://www.fuzzytech.com/binaries/ieccd1.pdf>.

ПРЕДМЕТНИЙ ПОКАЖЧИК

А

агрегування 102
активація 103
акумуляція 105, 106
алгоритми нечіткого висновку 116
аналіз кластерний 169, 171
антецедент 83, 89

Г

генетичний алгоритм 191

Д

дефазіфікація 110

З

зворотний зв'язк 39, 40

Е

egalt 212
евристика 33
етапи нечіткого висновку 116

К

композиція 89, 91
консеквент 89
кросовер 197

Л

лінгвістична змінна 57, 82
логічні операції 85, 88

М

машинне навчання 29
межі нечіткої множини 69
метод лівого модального
значення 113
метод правого модального
значення 114
метод центру ваги 111, 112, 115
метод центру площі 113, 115

методи створення ІІІ 26

методи управління 36

методологія НЛ 56

множина нечітка 32, 48, 66, 67

множина чітка 65

моделі системи з НЛК 51, 142

мутація 198

Н

невизначеність лінгвістична 61, 63

невизначеність стохастична 62, 64

невизначеність параметрична 15, 16

нейрон штучний 180, 181

нейронні мережі 15, 26, 177

нечітка змінна 82

нечітка логіка 15, 26, 32, 48

нечітке висловлювання 83, 84

нечіткі множини 32, 47, 66

носій НМ 67

О

обчислювальні технології 32

П

правила нечітких знань 89, 90

правило синтаксичне 58, 61

правило семантичне 58, 61

порожня НМ 67

Р

регулятори лінійні 13

С

секвенція 89

селекція 191, 193

синапс 179

синергетика 14, 15

системи електроенергетичні 13, 18, 2

структурна схема 35

схрещування 187, 197

Т

традиційні методи 13, 21, 47

У

універсум 67

І

інтелéкт 23

інтелектуальна мережа 16, 21

інтелектуальний контролер 40

Х

хромосома 192

Ф

фаззіфікація 99, 100

фітнес функція 192

функція активації 180

функція належності 58, 69

Ш

штúчний інтелéкт 23

Я

ядро нечіткої множини 68



КИРИК

Валерій Валентинович –

доктор технічних наук, професор,
завідувач кафедри електричних
мереж та систем Національного
технічного університету України
«Київський політехнічний інститут
імені Ігоря Сікорського».