ПРОЄКТ

Міністерство освіти і науки України
Донбаська державна машинобудівна академія (ДДМА)

О. В. Разживін,
О. В. Суботін

СИНТЕЗ НЕЧІТКИХ РЕГУЛЯТОРІВ В СИСТЕМАХ АВТОМАТИЧНОГО КЕРУВАННЯ

Навчальний посібник

Затверджено
на засіданні вченої ради
Протокол № від

Краматорськ
ЦТРІ «Друкарський дім»
2021

УДК 681.5 (075.8)
Р 7

Разживін, О. В.

Р 17 Синтез нечітких регуляторів в системах автоматичного керування: навчальний посібник / О. В. Разживін, О. В. Суботін. – Краматорськ : ЦТРІ «Друкарський дім», 2017. – 212 с.

ISBN

Викладено методики вирішення найбільш важливих завдань проектування автоматизованих систем керування з використанням нечіткої логіки.

УДК 681.5:621.77

ISBN

ЗМІСТ

ТЕМА 1 ОСНОВИ ТЕОРІЇ НЕЧІТКИХ МНОЖИН 4

ТЕМА 2. НЕЧІТКА ЛОГІКА 11

ТЕМА3 . НЕЧІТКІ ВІДНОШЕННЯ 25

ТЕМА 4 МОДЕЛЮВАННЯ В MATLAB 31

ТЕМА 5. НЕЧІТКА КЛАСТЕРИЗАЦІЯ 51

ТЕМА 6. РОЗВ'ЯЗУВАННЯ ЗАДАЧ В MATLAB 68

ТЕМА 7. НЕЙРО-НЕЧІТКІ МЕРЕЖІ 82

ТЕМА 8. БАЗИ ЗНАНЬ МАМДАНІ І СУДЖЕНО 95

ТЕМА 9. НЕЧІТКИЙ БАГАТОКРИТЕРАЛЬНИЙ АНАЛІЗ ВАРІАНТІВ 118

ТЕМА 10 АНАЛІЗ ВАРІАНТІВ «ЩО БУДЕ, ЯКЩО …» 134

ТЕМА 11. СИНТЕЗ НЕЧІТКИХ ДИНАМІЧНИХ МОДЕЛЕЙ У MATLAB 138

ТЕМА 12 РОЗРОБКА НЕЙРО-НЕЧІТКИХ МОДЕЛЕЙ. ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ У СЕРЕДОВИЩІ MATLAB 148

ТЕМА 13. ПРОЕКТУВАННЯ ІЄРАРХІЧНИХ НЕЧІТКИХ СИСТЕМ 164

ТЕМА 14. НЕЧІТКА ІЄРАРХІЧНА МОДЕЛЬ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ 189

ЛІТЕРАТУРА 208

ТЕМА 1 ОСНОВИ ТЕОРІЇ НЕЧІТКИХ МНОЖИН

Нечіткість знань. Розвиток теорії нечітких множин. Нечіткі множини та змінні. Функції приналежності. Основні типи параметричних функцій приналежності.

1.1 Нечіткість знань

При розробці інтелектуальних систем знання про конкретну предметну область, для якої створюється система, рідко бувають повними й абсолютно достовірними. Навіть кількісні дані, отримані шляхом досить точних експериментів, мають статистичні оцінки вірогідності, надійності, значимості і т.д. Інформація, якою заповнюються експертні системи, отримується у результаті опитування експертів, думки яких є суб'єктивними і можуть розходитися. Поряд із кількісними характеристиками в базах знань інтелектуальних систем повинні зберігатися якісні показники, евристичні правила, текстові знання і т.д. При обробці знань із застосуванням механізмів формальної логіки виникає протиріччя між нечіткими знаннями і чіткими методами логічного виведення. Розв'язати це протиріччя можна шляхом подолання нечіткості знань (коли це можливо) або використанням спеціальних методів подання й обробки нечітких знань.

Зміст терміна нечіткість багатозначний та включає такі основні компоненти: недетермінованість висновків, багатозначність інтерпретації, ненадійність знань і висновків, неповнота знань і немонотонна логіка, неточність знань.

Основні поняття нечіткості

Недетермінованість висновків – це характерна риса більшості систем штучного інтелекту. Недетермінованість означає, що заздалегідь шлях вирішення конкретної задачі в просторі її станів визначити неможливо. Тому в більшості випадків методом проб і помилок вибирається деякий ланцюжок логічних висновків, що узгоджуються з наявними знаннями, а у випадку якщо він не приводить до успіху, то організується перебір з поверненням для пошуку іншого ланцюжка і т. д. Такий підхід припускає визначення деякого первісного шляху. Недетермінованість висновків варто враховувати при розробці ефективних способів подання і збереження знань, а також при побудові методів пошуку й обробки знань, що дозволяють одержати рішення задачі за найменше число кроків. Для побудови таких методів звичайно застосовуються евристичні метазнання (знання про знання).

Багатозначність інтерпретації – звичайне явище в задачах розпізнавання. При розумінні природної мови серйозними проблемами стають багатозначність змісту слів, їхня підпорядкованість, порядок слів у реченні і т.п. Проблеми розуміння змісту виникають у будь-якій системі, що взаємодіє з користувачем природною мовою. Розпізнавання графічних образів також пов’язано з вирішенням проблеми багатозначної інтерпретації. При комп’ютерній обробці знань багатозначність необхідно усувати шляхом вибору правильної інтерпретації, для чого розроблено спеціальні методи.

Ненадійність знань і висновків означає, що для оцінки вірогідності знань не можна застосувати двобальну шкалу (1 – абсолютно достовірні знання, 0 – недостовірні знання). Для більш тонкої оцінки вірогідності знань застосовується імовірнісний підхід, заснований на теоремі Байєса, і інші методи (наприклад, метод висновків з використанням коефіцієнтів упевненості). Широке застосування на практиці одержали нечіткі висновки, які будуються на базі нечіткої логіки, що веде своє походження від теорії нечітких множин.

Неповнота знань і немонотонна логіка. Абсолютно повних знань не буває, оскільки процес пізнання нескінченний. У зв’язку з цим стан бази знань повинний змінюватися з часом. На відміну від простого додавання інформації, як у базах даних, при додаванні нових знань виникає небезпека одержання суперечливих висновків: тобто висновки, отримані з використанням нових знань, можуть спростовувати ті, що були отримані раніше. Ще гірше, якщо нові знання будуть знаходитися в протиріччі з старими, тоді механізм висновку може стати непрацездатним.

Більшість експертних систем першого покоління були засновані на моделі закритого світу, обумовленій застосуванням апарату формальної логіки для обробки знань. *Модель закритого світу* припускає твердий добір знань, що включаються в базу, а саме: база знань заповнюється винятково вірними поняттями, а усе, що ненадійно або невиразно, свідомо вважається помилковим. Така модель має обмежені можливості подання знань і таїть у собі небезпеку одержання протиріччя при додаванні нової інформації. Недоліки моделі закритого світу пов'язані з тим, що формальна логіка виходить з передумови, відповідно до якої набір визначених у системі аксіом (знань) є *повним* (теорія є повною, якщо кожний її факт можна довести, виходячи з аксіом цієї теорії). Для повного набору знань справедливість раніше отриманих висновків не порушується з додаванням нових фактів. Ця властивість логічних висновків називається *монотонністю*. На жаль, реальні знання, що закладаються в експертні системи, украй рідко бувають повними.

Неточність знань. Відомо, що кількісні дані (знання) можуть бути неточними, при цьому існують кількісні оцінки такої неточності (довірчий інтервал, рівень значимості, ступінь адекватності і т.д.). Лінгвістичні знання також можуть бути неточними. Для врахування неточності лінгвістичних знань використовується теорія нечітких множин. Фактично нечіткість може бути ключем до розуміння здатності людини справлятися з задачами, що занадто складні для вирішення на ЕОМ. Розвиток досліджень в області нечіткої математики призвів до появи нечіткої логіки і нечітких висновків, що виконуються з використанням знань, представлених нечіткими множинами, нечіткими відношеннями, нечіткими відповідностями і т. д.

1.2 Розвиток теорії нечітких множин

Теорія нечітких множин (fuzzy sets theory) бере свій початок з 1965 року, коли професор Лотфі Заде (Lotfi Zadeh) з університету Берклі в США опублікував основну роботу «Fuzzy Sets» у журналі «Information and Control». Прикметник «fuzzy» (нечіткий, розмитий), введено в назву нової теорії з метою відокремлення від традиційної чіткої математики й аристотелевої логіки, що оперують з чіткими поняттями: «належить – не належить», «істина – хибність». Концепція нечіткої множини зародилася у Заде «як незадоволеність математичними методами класичної теорії систем, що змушувала домагатися штучної точності, недоречної в багатьох системах реального світу, особливо в так званих гуманістичних системах, що включають людей».

Початком практичного застосування теорії нечітких множин можна вважати 1975 рік, коли Е. Мамдані (E. Mamdani) побудував перший нечіткий контролер. Успіх першого промислового контролера, заснованого на нечітких лінгвістичних правилах «Якщо – то» привів до сплеску інтересу до теорії нечітких множин серед математиків та інженерів.

Досягнення науковців.

Можливість використання нечіткої логіки базується на таких результатах.

У 1992 р. Коско (B.Kosko) була доведена *теорема про нечітку апроксимацію* (Fuzzy Approximation Theorem), відповідно до якої будь-яка матема- тична система може бути апроксимована системою, заснованою на нечіткій логіці. Іншими словами, за допомогою природно-мовних висловлень-правил «Якщо – то», з подальшою їх формалізацією засобами теорії нечітких множин, можна скільки завгодно точно відбити довільний взаємозв'язок «вхід – вихід» без використання складного апарата диференціального й інтегрального числень, традиційно застосовуваного в керуванні й ідентифікації.

У 1992 р. Ванг (L.Wang) показав, що нечітка система є універсальним апроксиматором, тобто може апроксимувати будь-яку неперервну функцію з довільною точністю, якщо використовує набір з n ($n \to \infty$) правил виду «Якщо – то», гаусові функції приналежності, композиції у вигляді добутку, імплікації у формі Ларсена та центроїдний метод приведення до чіткості.

У 1995 р. Кастро (J. Castro) показав, що логічний контролер Мамдані також є універсальним апроксиматором при симетричних трикутних функціях приналежності, композиції з використанням операції мінімум, імплікації у формі Мамдани і центроїдного методу приведення до чіткості.

Системи з нечіткою логікою *доцільно застосовувати* для складних процесів, коли немає простої математичної моделі, а також якщо експертні знання про об'єкт або про процес можна сформулювати тільки в лінгвістичній формі.

Системи, що базуються на нечіткій логіці, *застосовувати недоцільно* якщо необхідний результат може бути отриманий яким-небудь іншим (стандартним) шляхом, або якщо для об'єкта або процесу вже знайдена адекватна і легко досліджувана математична модель.

Основні недоліки систем з нечіткою логікою:

1) вихідний набір нечітких правил, що постулюються, формулюється експертом-людиною і може виявитися неповним або суперечливим;

2) вид і параметри функцій приналежності, що описують вхідні і вихідні змінні системи, вибираються суб'єктивно і мо жуть виявитися такими, що цілком не відбивають реальну дійсність.

1.3 Нечіткі множини та змінні

Нехай U – універсальна множина, u – елемент U, а G – деяка властивість. *Звичайна (чітка) підмножина A* універсальної множини U, елементи якої мають властивість G, визначається як множина впорядкованих пар $\{<\mu_A(u)|u>\}$, де $\mu_A(u)$ – характеристична функція приналежності, що приймає значення 1, якщо u має властивість G, та 0 – у протилежному випадку.

Нечітка підмножина відрізняється від звичайної тим, що для елементів u з U немає однозначної відповіді «ні» або «так» щодо властивості G.

У зв'язку з цим *нечітка підмножина A* універсальної множини U визначається як множина впорядкованих пар А = $\{<\mu_A(u)|u>\}$, де $\mu_A(u)$ –

характеристична *функція приналежності* (або просто функція приналежності), що приймає значення в деякій цілком впорядкованій множині M (наприклад, M = [0; 1]).

1.4 Функція приналежності вказує ступінь приналежності елемента u нечіткій підмножині A. Множину M називають *множиною приналежностей*. Якщо M={0, 1}, то нечітка підмножина А може розглядатися як чітка множина. Чітку множину A можна розглядати як граничний випадок нечіткої множини А, функція приналежності якої $\mu_A(u)$ набуває лише бінарних значень.

Приклади

Приклад 1.1. Представити у вигляді нечіткої множини поняття «Чоловік середнього зросту») на універсальній множині U={155,160,165,170,175,180,185,190}. Одне з можливих рішень виглядає так:

A=(0/155, 0,1/160, 0,3/165, 0,8/170, 1/175, 1/180, 0,5/185, 0/190).

Нечітка змінна визначається як $<a, U, A>$, де a – найменування змінної, $U=\{u\}$ – область визначення змінної (набір можливих значень u), $A=\{<\mu_A(u)|u>\}$ – нечітка множина, що описує обмеження на можливі значення змінної a (семантику).

Нечітка змінна – це теж саме, що і нечітке число, тільки з додаванням імені, яким формалізується поняття, що описується цим числом. Для людини зручніше задавати значення змінної не числами, а словами. Щодня ми приймаємо рішення на основі лінгвістичної інформації типу: «дуже висока температура»; «утомлива поїздка»; «швидка відповідь»; «красивий букет»; «гармонійний смак» і тому подібне. Психологи встановили, що в людському мозку майже вся числова інформація вербально перекодується і зберігається у вигляді слів.

Лінгвістична змінна – це множина нечітких змінних, вона використовується для того, щоб дати словесний опис деякому нечіткому числу, отриманому в результаті деяких операцій.

Лінгвістична змінна визначається як $<x, L, U, G, M>$, де x – найменування змінної, L – множина її значень (базова терм-множина), що складається з найменувань нечітких змінних, областю визначення кожної з яких є множина U; G – синтаксична процедура (граматика), що дозволяє оперувати елементами терм-множини L, зокрема – генерувати нові осмислені терми; *L*'=*L*È*G*(*L*) задає розширену терм-множину (È – знак об'єднання); M – семантична процедура, що дозволяє приписати кожному новому значенню лінгвістичної змінної нечітку семантику, шляхом формування нової нечіткої множини.

Терм-множина – це множина всіх можливих значень лінгвістичної змінної.

Терм – будь-який елемент терм-множини. У теорії нечітких множин терм формалізується нечіткою множиною за допомогою

функції приналежності.

Наприклад, змінна «швидкість автомобіля» може набувати значень «низька», «середня», «висока» і «дуже висока». В цьому випадку лінгвістичною змінною є «швидкість автомобіля», термами – лінгвістичні оцінки «низька», «середня», «висока» і «дуже висока», які і складають терм-множину.

Нечіткий терм – це нечітка множина, яка має властивість, якій відповідає певне поняття.

ТЕМА 2. НЕЧІТКА ЛОГІКА

Лінгвістичні змінні

Нагадаємо, що **лінгвістичною називається змінна**, що приймає значення з множини слів або словосполук деякої природної мови. Поняття лінгвістичної змінної відіграє важливу роль в нечіткому логічному виведенні та в ухваленні рішень на основі наближених міркувань. Формально лінгвістична змінна описується наступною п'ятіркою:

$<x, T, U, G, M>$,

де x – ім'я змінної; T - терм-множина, кожен елемент якої задається нечіткою множиною на універсальній множині U; G - синтаксичні правила (часто у вигляді граматики), що породжують назву термів; M – семантичні правила, що задають функції приналежності нечітки х термів, породжених синтаксичними правилами з G.

Приклад 1: Розглянемо лінгвістичну змінну з ім'ям x = «температура в кімнаті». Тоді четвірку $< T, U, G, M>$, що залишилася, можна визначити так:

· універсальна множина: U=[12, 35];

· терм-множина: T= {«холодно», «комфортно», «спекотно»} з такими функціями приналежності:

$$\mu_{«холодно»}(u)=\frac{1}{1+\left|\frac{u-12}{6}\right|^{12}};\ \mu_{«комфортно»}(u)=\frac{1}{1+\left|\frac{u-20}{3}\right|^{8}};\ \mu_{«жарко»}(u)=\frac{1}{1+\left|\frac{u-33}{8}\right|^{12}};\ u\in U;$$

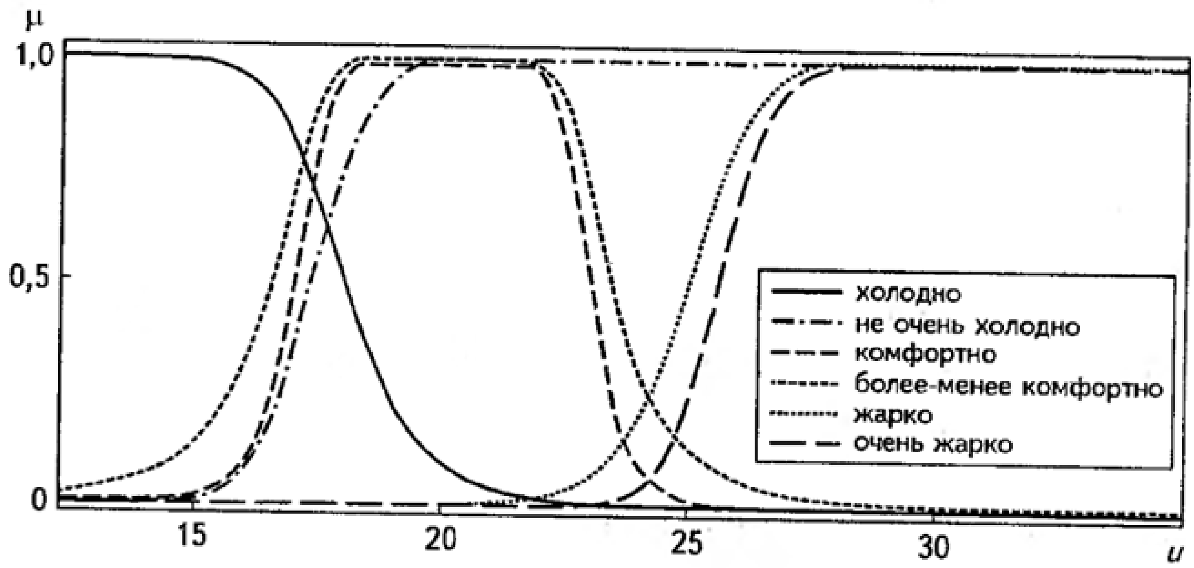
· синтаксичні правила G, що породжують нові терми з використанням квантифікаторів «не», «дуже» і «більш-менш»;

· семантичні правила М, задані в табл.1.

Таблиця 1 – Правила модифікації функцій приналежності

Квантификатор	Функция принадлежности
Не t	$1-\mu_t(u)$
Очень t	$(\mu_t(u))^2$
Более-менее t	$\sqrt{\mu_t(u)}$

Графіки функцій приналежності термів «холодно», «не дуже холодно», «комфортно», «більш-менш комфортно», «спекотно» і «дуже спекотно» лінгвістичною змінною «температура в кімнаті» показані на рис. 1.



Приклад 2: Експерт визначає товщину виробу, що випускається, за допомогою понять «Мала товщина», «Середня товщина» і «Велика товщина», при цьому мінімальна товщина дорівнює 10 мм, а максимальна — 80 мм.

Формалізація такого опису може бути проведена за допомогою наступної лінгвістичної змінної (x, T, U, G, M), де

x – товщина виробу;

T – {«Мала товщина», «Середня товщина», «Велика товщина»};

U = [10,80];

G – процедура створення нових термів за допомогою зв'язок «та», «або» і квантифікаторів типа «дуже», «не», «злегка» і тому подібне. Наприклад: «мала або середня товщина», «дуже мала товщина» і т.д.;

M – процедура завдання на U=[10, 80] нечітких підмножин A_1 = «Мала товщина», A_2— «Середня товщина», A_3 = «Велика товщина», а також нечітких множин для термів з $G(T)$ відповідно до правил трансляції нечітких зв’язок і квантифікаторів «та», «або», «не», «дуже», «злегка» і інших операцій над нечіткими множинами виду: *A*Ç*B*, *A*È*B*, , CON A=A^2, DIL $A^{0,5}$ і т.

Терм-множину і розширену терм-множину в умовах прикладу можна охарактеризувати функціями приналежності, приведеними на рис. 2 і 3.

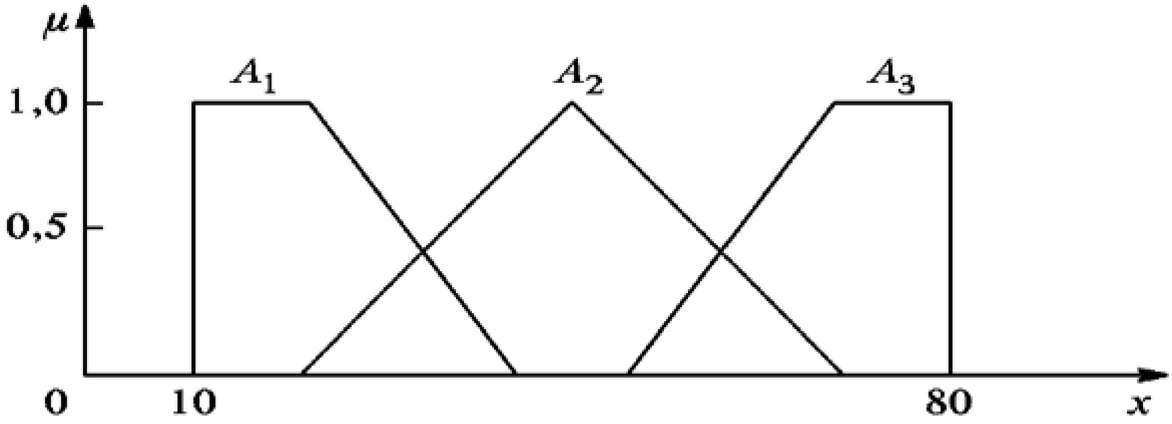


Рис. 2 – Функції приналежності нечітких множин:

«Мала товщина» – A_1, «Середня товщина» – A_2, «Велика товщина» – A_3

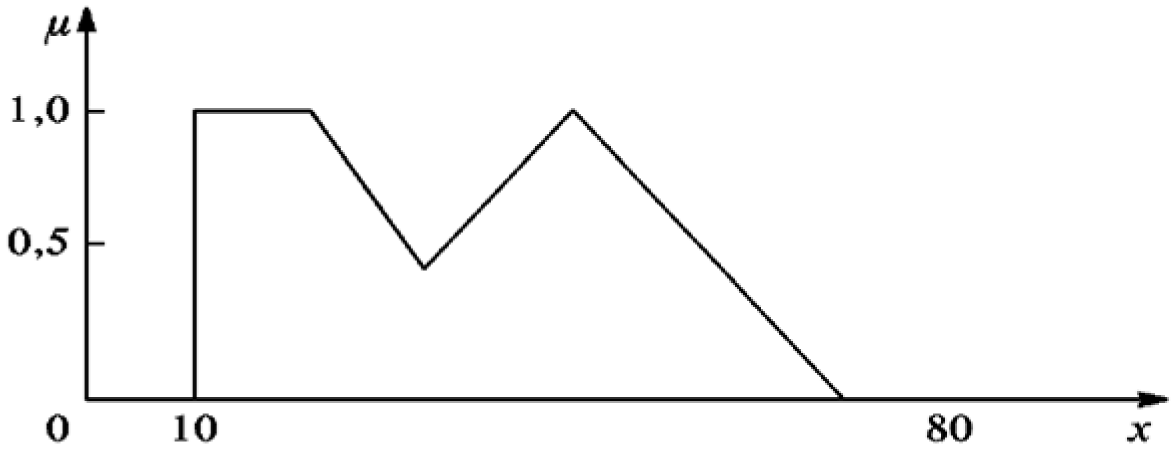


Рис. 3 – Функція приналежності нечіткої множини:

«Мала або середня товщина» – A_1ÈA_3

Нечітка істинність

Особливе місце в нечіткій логіці займає **лінгвістична змінна «істинність»**. У класичній логіці істинність може набувати лише два значення: *істинно і помилково*. У нечіткій логіці істинність «розмита». Нечітка істинність визначається аксиоматично, причому різні автори виконують це по різному. Інтервал [0,1] використовується як універсальна множина для лінгвістичної змінної «істинність». Звичайна, чітка істинність може бути представлена нечіткими множествами-синглтонами. В цьому випадку чітким поняттям «істинно» і «помилково» відповідатимуть такі функції приналежності:

$$\mu_{«истинно»}(u) = \begin{cases} 0, & u \neq 1 \\ 1, & u = 1 \end{cases} \quad и \quad \mu_{«ложно»}(u) = \begin{cases} 0, & u \neq 0 \\ 1, & u = 0; \end{cases} \quad u \in [0,\ 1].$$

Для нечіткої істинності **Заді** запропонував наступні функції приналежності термів «істинно» і «помилково»:

$$\mu_{«истинно»}(u) = \begin{cases} 0, & 0 \le u \le a \\ 2\left(\dfrac{u-a}{1-a}\right)^2, & a < u \le \dfrac{a+1}{2}; \\ 1-2\left(\dfrac{u-1}{1-a}\right)^2, & \dfrac{a+1}{2} < u \le 1 \end{cases} \quad \mu_{«ложно»}(u) = \mu_{«истинно»}(1-u); \ u \in [0,\ 1],$$

де aÎ[0,1] – параметр, що задає носії нечітких множин «істинно» і «помилково». Для нечіткої множини «істинно» носієм буде інтервал (a, 1], а для нечіткої множини «помилково» – [0, a). Графіки функцій приналежності нечітких термів «істинно» і «помилково» при a=0,4 представлені на рис.1. Вони є дзеркальними відображеннями.

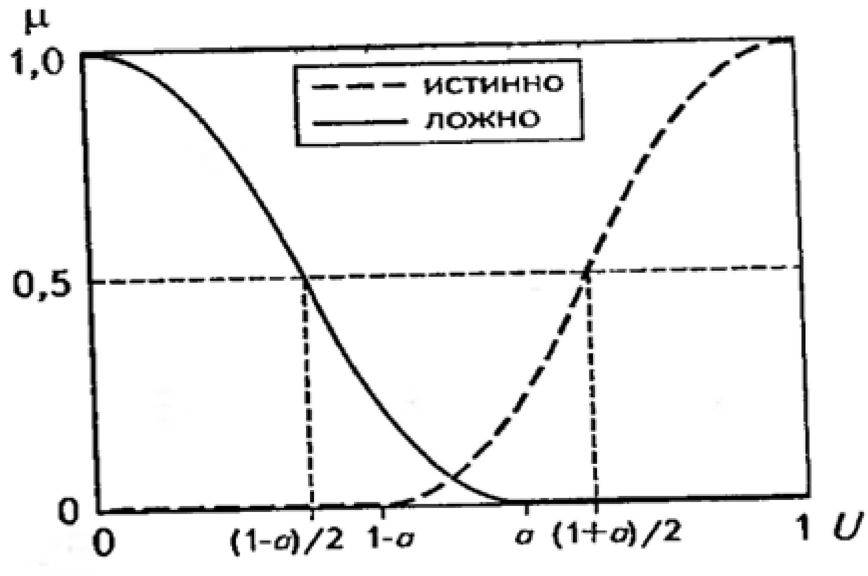


Рис.1 – Лінгвістична змінна «істинність» за Заде

Для нечітких значень «істинно» і «помилково» Балдвін запропонував такі функції приналежності:

$$\mu_{«истинно»}(u) = u\,;\ \ \mu_{«ложно»}(u) = 1 - u\,;\ \ u \in [0,\ 1].$$

Квантіфікатори «більш-менш» і «дуже» застосовують до нечітких значень «істинно» і «помилково», отримуючи терми «дуже помилково», «більш-менш помилково», «більш-менш істинно», «дуже істинно», «дуже, дуже істинно», «дуже, дуже помилково» і тому подібне. Функції приналежності нових термів розраховують з використанням операцій концентрації і розтягування нечіткої множини, що відповідає піднесенню функції приналежності до ступеня 2 і 1/2 відповідно:

$$\mu_{«очень\ ложно»}(u) = (\ldots\mu_{«ложно»}(u))^2\,;$$
$$\mu_{«очень,\ очень\ ложно»}(u) = (\ldots\mu_{«очень\ ложно»}(u))^2\,;$$
$$\mu_{«более-менее\ ложно»}(u) = (\ldots\mu_{«ложно»}(u))^{1/2}\,;$$
$$\mu_{«более-менее\ истинно»}(u) = (\ldots\mu_{«истинно»}(u))^{1/2}\,;$$
$$\mu_{«очень\ истинно»}(u) = (\ldots\mu_{«истинно»}(u))^2\,;$$
$$\mu_{«очень,\ очень\ истинно»}(u) = (\ldots\mu_{«очень\ истинно»}(u))^2\,.$$

Графіки цих функцій приналежності показані на рис.2.

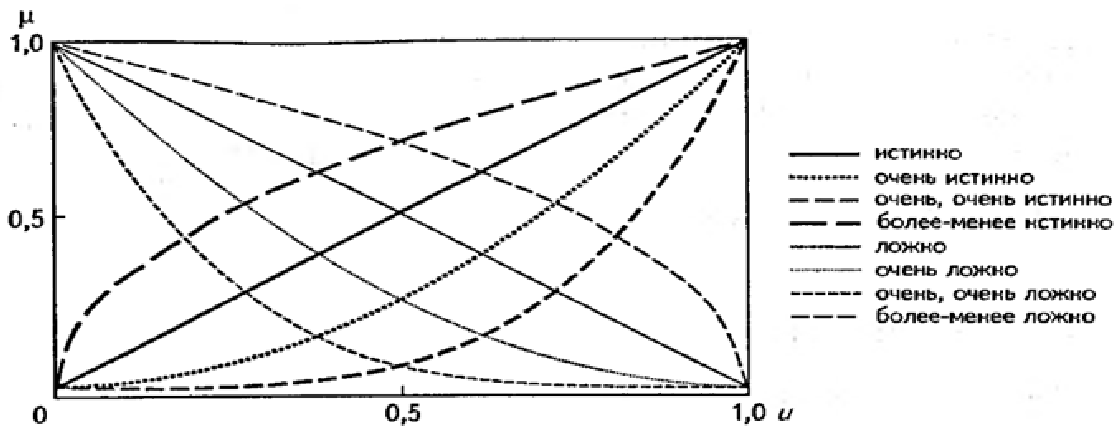


Рис.2 – Лінгвістична змінна «істинність» за Балдвіном

Процес і система нечіткого логічного виведення

Використовуваний в різного роду **експертних та керуючих системах** механізм нечіткого виведення в своїй основі має **базу знань**, що формується фахівцями предметної області у вигляді сукупності нечітких предикативних правил вигляду:

$$П_1: \text{якщо } x \in A_1, \text{ тоді } y \in B_1,$$

$$П_2: \text{якщо } x \in A_2, \text{ тоді } y \in B_2,$$

$$\dots$$

$$П_n: \text{якщо } x \in A_n, \text{ тоді } y \in B_n,$$

де x – вхідна змінна (ім’я для відомих значень даних), y – змінна виведення (ім’я для значення даних, яке буде обчислено); A і B – функції приналежності, визначені відповідно на x і y.

Приклад подібного правила

Якщо x – низько, то y – високо.

Приведемо детальніше пояснення. Знання експерта A ® B відображає нечітке причинне відношення передумови і висновку, тому його можна назвати нечітким відношенням і позначити через R:

$R = A$ ® B.

де «®» називають ***нечіткою імплікацією***.

Відношення R можна розглядати як нечітку підмножину прямого добутку $X´Y$ повної множини передумов X і висновків Y. Таким чином, процес здобуття (нечіткого) результату висновку з використанням даного спостереження ; і знання A®B можна представити у вигляді формули

= °R= °(A®B),

де ° – **операція згортки** (композиції).

Як операцію композиції, так і операцію імплікації в алгебрі нечітких множин можна реалізовувати по-різному (при цьому, природно, різнитиметься і підсумковий отримуваний результат), але у будь-якому випадку ***загальне логічне виведення*** здійснюється за наступні чотири етапи.

1. ***Нечіткість*** (введення нечіткості, фазифікація, fuzzification). Функції приналежності, визначені на вхідних змінних, застосовуються до їх фактичних значень для визначення міри істинності кожної передумови кожного правила.

2. ***Логічне виведення.*** Обчислене значення істинності для передумов кожного правила застосовується до висновків кожного правила. Це призводить до однієї нечіткої підмножини, яка буде призначена кожній змінній виведення для кожного правила. Як правила логічного висновку зазвичай використовуються лише операції min (МІНІМУМ) або prod (МНОЖЕННЯ). У логічному виведенні МІНІМУМУ функція приналежності виведення «відсікається» по висоті, відповідній обчисленій мірі істинності передумови правила (нечітка логіка «ТА»). У логічному виведенні МНОЖЕННЯ функція приналежності виводу масштабується за допомогою обчисленої міри істинності передумови правила.

3. ***Композиція.*** Всі нечіткі підмножини, призначені до кожної змінної виведення (у всіх правилах), об'єднуються разом, аби сформувати одну нечітку підмножину для кожної змінної виведення. При подібному об'єднанні зазвичай використовуються операції max (МАКСИМУМ) або sum (СУМА). При композиції МАКСИМУМУ комбіноване виведення нечіткої підмножини конструюється як поточковий максимум по всіх нечітких підмножинах (нечітка логіка «АБО»). При композиції СУМИ комбіноване виведення нечіткої підмножини конструюється як поточкова сума по всіх нечітких підмножинах, призначених змінній виведення правилами логічного виводу.

4. На закінчення (додатково) – ***приведення до чіткості*** (дефазифікація, defuzzification), яке використовується, коли корисно перетворити нечіткий набір висновків в чітке число. Є велика кількість методів приведення до чіткості, деякі з яких розглянуті нижче.

Приклад : Хай деяка система описується наступними нечіткими правилами:

$П_1$: якщо x є A, то w є D;

$П_2$: якщо y є B, то w є E;

$П_3$: якщо z є C, то w є F,

де x, y, z – імена вхідних змінних, w – ім'я змінної виведення, а A, B, C, D, E, F – задані функції приналежності (трикутної форми).

Процедура здобуття логічного висновку ілюструється рис.1.

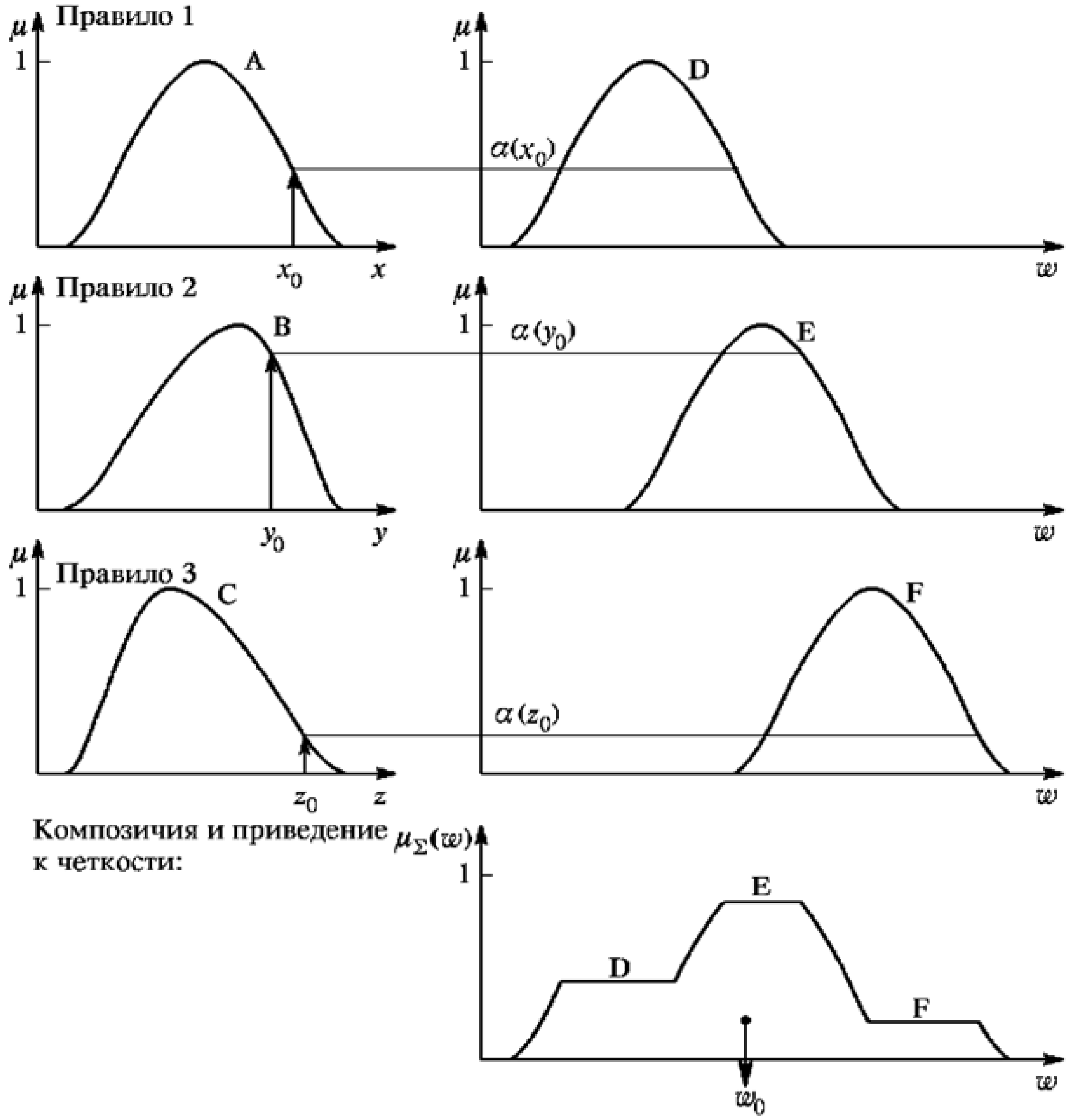


Рис.1 – Ілюстрація до процедури логічного висновку

Передбачається, що вхідні змінні набули деяких конкретних (чітких) значень — x_0, y_0, z_0.

Відповідно до приведених етапів, на етапі 1 для даних значень і виходячи з функцій приналежності A, B, C, знаходяться міри істинності $a(x_0)$, $a(y_0)$ і $a(z_0)$ для передумов кожного з трьох приведених правил (див.рис.4.4).

На етапі 2 відбувається «відсікання» функцій приналежності висновків правил (тобто D, E, F) на рівнях $a(x_0)$, $a(y_0)$ і $a(z_0)$.

На етапі 3 розглядаються усічені на другому етапі функції приналежності і виконується їх об'єднання з використанням операції max, внаслідок чого виходить комбінована нечітка підмножина, що описується функцією приналежності і відповідає логічному висновку для вихідної змінної *w*.

Нарешті, на 4-му етапі – при необхідності знаходиться чітке значення вихідної змінної, наприклад, із застосуванням центроїдного методу: чітке значення вихідної змінної визначається як центр тяжіння для кривої , тобто:

$$w_0 = \frac{\int\limits_{\Omega} w\mu_{\Sigma}(w)\,dw}{\int\limits_{\Omega} \mu_{\Sigma}(w)\,dw}.$$

Система нечіткого виведення складається з п'яти функціональних блоків (рис.2):

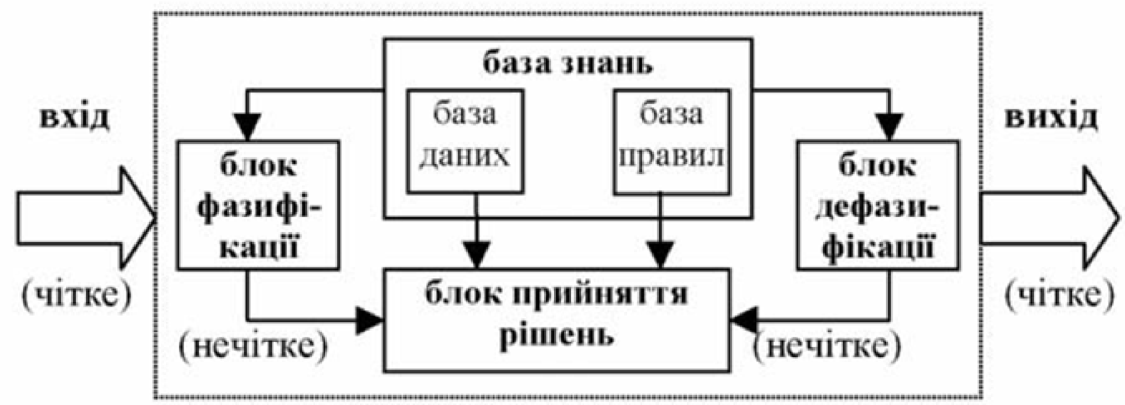


Рис.2 – Система нечіткого виведення

– ***блок фазифікації***, що перетворює чисельні вхідні значення в ступінь відповідності лінгвістичним змінним;

– ***база правил***, що містить набір нечітких правил типу якщо-то;

– ***база даних***, у якій визначені функції приналежності нечітких множин, що використовуються в нечітких правилах;

– ***блок прийняття рішень***, який виконує операції виведення на основі існуючих правил;

– ***блок дефазифікації***, що перетворює результати виведення в чисельні значення.

Виділяють три основних ***типи систем нечіткого виведення***:

– **1-й тип**: вихідне значення знаходиться як зважене середнє результатів виконання кожного правила, для кожного з яких дефазифікація проводиться окремо; для таких систем вихідні функції приналежності повинні бути монотонно-неспадаючими;

– **2-й тип**: вихідне нечітке значення – це результат об'єднання нечітких виходів кожного правила; кожний нечіткий вихід зважено за допомогою ваг спрацьовування правил; чітке вихідне значення знаходиться в результаті дефазифікації об'єднаного нечіткого виходу;

– **3-й тип**: система, побудована на правилах типа Сугено; вихідне значення є лінійною комбінацією вхідних значень плюс деяке постійне значення, загальний вихід є середнім зваженим всіх правил.

В загальному випадку в якості значень вхідних та вихідних змінних правил можна використовувати нечіткі множини, з якими не пов'язано ніяке поняття – оскільки при проведенні нечіткого виведення нечіткі терми все одно представляються нечіткими множинами і пов'язане з нечітким термом поняття не відіграє ніякої ролі.

Фазифікація (fuzzification) – це визначення ступеня виконання антецедентів правил. За допомогою фазифікації чіткому значенню ставляться у відповідність ступені його приналежності до нечітких множин.

Дефазифікація (defuzzification) – процедура перетворення нечіткої множини в чітке число за ступенем приналежності.

В системах нечіткого виведення функції консеквенти, отримані в результаті виконання правил, об'єднуються в одну функцію $\mu(y)$. Існують різні методи дефазифікації цієї об'єднаної функції приналежності.

Нехай y – нечітка змінна, Y – область визначення змінної y, y^* – чітке значення нечіткої змінної y.

Методи нечіткого виведення

Розглянемо наступні найбільш часто використовувані модифікації алгоритму нечіткого виведення, вважаючи, для простоти, що базу знань організовують два нечітких правила вигляду:

$$П_1: \text{якщо } x \in A_1 \text{ і } y \in B_1, \text{ то } z \in C_1;$$

$$П_2: \text{якщо } x \in A_2 \text{ і } y \in B_2, \text{ то } z \in C_1,$$

де x, y – імена вхідних змінних, z – ім'я змінної виведення, а A_1, B_1, C_1 – задані функції приналежності, при цьому чітке значення z_0 необхідно визначити ні основі приведеної інформації та чітких значень x_0 і y_0 .

Алгоритм Mamdani. Даний алгоритм відповідає розглянутому прикладу 4.3 і рис.4.4. У даній ситуації він математично може бути описаний таким чином.

1. ***Нечіткість***: знаходяться міри істинності для передумов кожного правила: $A_1(x_0)$, $A_2(x_0)$, $B_1(y_0)$, $B_2(y_0)$.

2. ***Нечітке виведення***: знаходяться рівні «відсікання» для передумов кожного з правил (з використанням операції МІНІМУМ)

$$\alpha_1 = A_1(x_0) \wedge B_1(y_0),$$
$$\alpha_2 = A_2(x_0) \wedge B_2(y_0),$$

де через «Ù» позначена операція логічного мінімуму (min), потім знаходяться «усічені» функції приналежності

$$C_1'(z) = (\alpha_1 \wedge C_1(z)),$$
$$C_2'(z) = (\alpha_2 \wedge C_2(z)).$$

3. ***Композиція***: з використанням операції МАКСИМУМ (max, що далі позначається як «Ú») виконується об'єднання знайдених усічених функцій, що призводить до здобуття підсумкової нечіткої підмножини для змінної виведення з функцією приналежності

$$\mu_\Sigma(z) = C(z) = C_1'(z) \vee C_2'(z) = (\alpha_1 \wedge C_1(z)) \vee (\alpha_2 \wedge C_2(z)).$$

4. Нарешті, приведення до чіткості (для знаходження z_0) проводиться, наприклад, методом центру тяжіння (центроїдним).

Алгоритм Sugeno. Sugeno і Takagi використовували набір правил в наступній формі (як і раніше, наводиться приклад з двох правил):

$П_1$: якщо x є A_1 і y є B_1, то $z_1=a_1x+b_1y$;

$П_2$: якщо x є A_2 і y є B_2, то $z_2=a_2x+b_2y$.

Представлення алгоритму

1. Перший етап – як в алгоритмі Mamdani.

2. На другому етапі знаходяться

$$\alpha_1 = A_1(x_0) \wedge B_1(y_0), \ \alpha_2 = A_2(x_0) \wedge B_2(y_0)$$

та індивідуальні виходи правил

$$z_1^* = a_1x_0 + b_1y_0,$$
$$z_2^* = a_2x_0 + b_2y_0.$$

3. На третьому етапі визначається чітке значення змінної виведення:

$$z_0 = \frac{\alpha_1 z_1^* + \alpha_2 z_2^*}{\alpha_1 + \alpha_2}.$$

Ілюструє алгоритм рис.6.

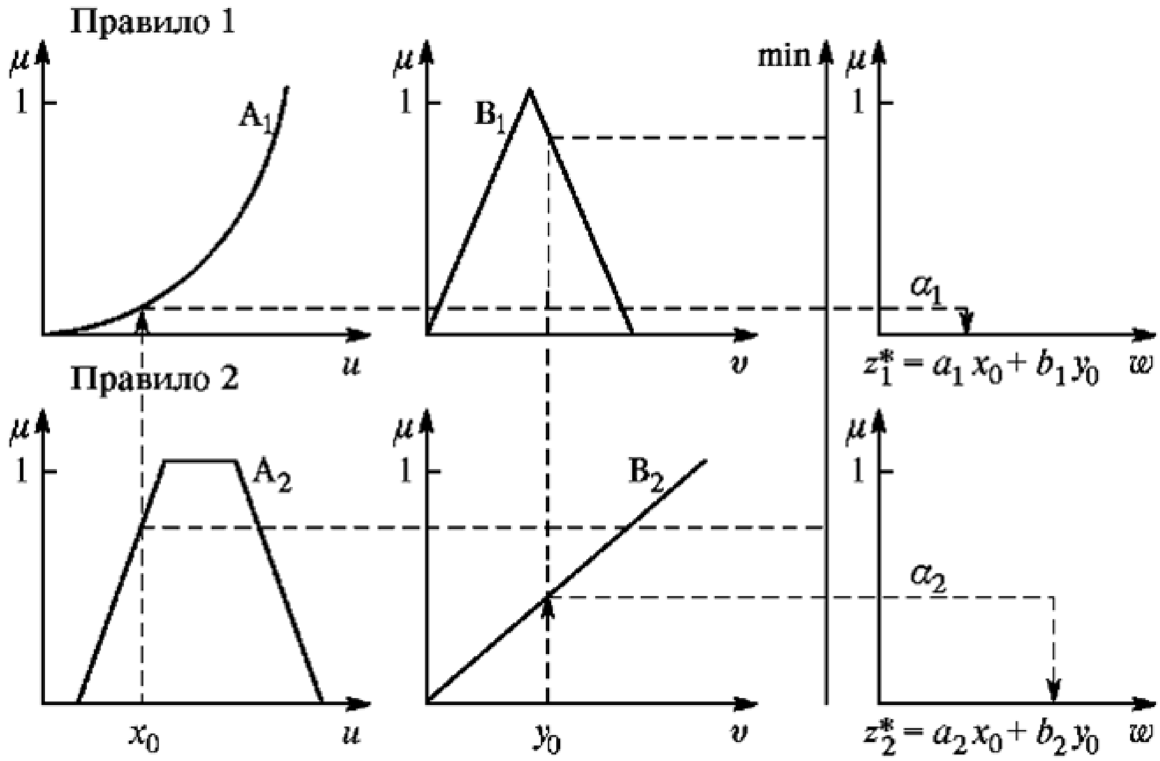


Рис.6 – Ілюстрація до алгоритму Sugeno

Алгоритм Larsen. У алгоритмі Larsen нечітка імплікація моделюється з використанням оператора множення.

Опис алгоритму

1. Перший етап – як в алгоритмі Mamdani.

2. На другому етапі, як в алгоритмі Mamdani спочатку знаходяться значення

$$\alpha_1 = \mathrm{A}_1(x_0) \wedge \mathrm{B}_1(y_0),$$
$$\alpha_2 = \mathrm{A}_2(x_0) \wedge \mathrm{B}_2(y_0),$$

а потім часткові нечіткі підмножини

$$\alpha_1 C_1(z), \quad \alpha_2 C_2(z).$$

3. Знаходиться підсумкова нечітка підмножина з функцією приналежності

$$\mu_\Sigma(z) = C(z) = (\alpha_1 C_1(z)) \vee (\alpha_2 C_2(z))$$

У загальному випадку *n* правил

$$\mu_\Sigma(z) = C(z) = \bigvee_{i=1}^{n} (\alpha_i C_i(z)$$

4. При необхідності виконується приведення до чіткості (як в раніше розглянутих алгоритмах).

Алгоритм Larsen ілюструється рис.7.

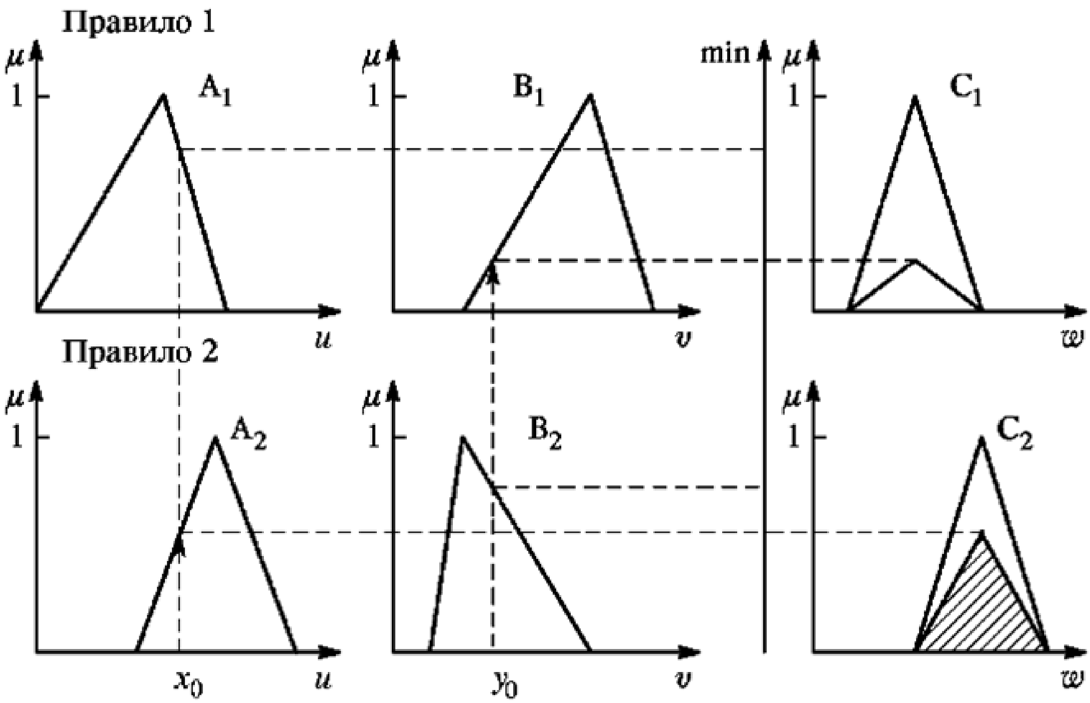


Рис.7 – Ілюстрація до алгоритму Larsen

Додатково:

https://uk.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D0%BD%D0%B5%D1%87%D1%96%D1%82%D0%BA%D0%BE%D0%B3%D0%BE_%D0%B2%D0%B8%D0%B2%D0%B5%D0%B4%D0%B5%D0%BD%D0%BD%D1%8F

ТЕМА3 . НЕЧІТКІ ВІДНОШЕННЯ

Нечіткі відношення

Нечітке n-арне відношення визначається як нечітка підмножина R на E, що приймає свої значення в M, де $E=E_1\times E_2\times\ldots\times E_n$ – прямий добуток універсальних множин, M – деяка множина приналежностей (наприклад, $M = [0; 1]$).

У випадку n=2 і $M = [0, 1]$ ***бінарним нечітким відношенням*** R між множинами $X = E_1$ і $Y = E_2$ буде називатися функція R: $(X, Y)\rightarrow[0, 1]$, що ставить у відповідність кожній парі елементів $(x, y)\in X\times Y$ величину $\mu_R(x, y) \in [0; 1]$, яка показує рівень виконання відношення R між елементами $(x, y) \in X\times Y$.

Нечітке відношення на $X\times Y$ записується у вигляді: $x\in X$, $y\in Y$: $x\,R\,y$.

У випадку, коли $X = Y$, тобто X і Y збігаються, нечітке відношення R: $X\times X\rightarrow[0,1]$ називається нечітким відношенням на множині X.

Порожнім нечітким відношенням $\varnothing$ називають відношення, що не містить жодного ***кортежу*** – довільного набору впорядкованих елементів.

Повним нечітким відношенням є декартовий добуток універсумів

$E = E_1 \times E_2 \times \ldots \times E_n$.

Способи задавання нечітких відношень використовують такі:

– у формі списку з явним перерахуванням усіх кортежів нечіткого відношення та відповідних ним значень функції приналежності:

$R=\{(w_1, \mu_R(w_1)), \ldots, (w_r, \mu_R(w_r))\}$,

де $w_i = \langle x_1, x_2, \ldots, x_n\rangle$ – i-ий кортеж елементів цього відношення, а r – число кортежів нечіткого відношення R;

– ***аналітично*** у формі певного математичного виразу для відповідної функції приналежності цього відношення.

Нечітке бінарне відношення може бути подане:

– ***графічно*** у вигляді певної поверхні або сукупності окремих точок у тривимірному просторі, при цьому вісі абсциси та ординати будуть відповідати універсумам E_1 та E_2, а вісь аплікати – інтервалу [0;1];

– ***у матричній формі*:** строки матриці нечіткого відношення при цьому відповідають першим, а стовпці – другим елементам кортежів, елементами матриці є відповідні значення функції приналежності нечіткого відношення;

– ***орієнтованим нечітким графом*** G = (V, E, μ_G), що може бути заданий у вигляді двох звичайних скінчених множин: множини вершин нечіткого графа V={v_1, v_2, …, v_n} та множини дуг нечіткого графа E={e_1, e_2, …, e_m}, а також певної функції приналежності дуг даному нечіткому графу μ_G : E →[0;1]. Вершинами нечіткого графу є елементи множин X і Y. Міра приналежності μ_R(x, y) показує наскільки сильно вершини xÎX і yÎY пов’язані між собою.

Нечіткий предикат P(<x_1, x_2, …, x_n>) – деяке відображення з декартового добутку універсумів E_1´E_2´ ... ´E_n у певну цілковито впорядковану множину значень істинності, зокрема, у інтервал [0;1]. При цьому змінні x_1, x_2, …, x_n називають предметними змінними нечіткого предиката, а декартовий добуток універсумів E_1´E_2´. ... ´E_n – предметною областю нечіткого предиката.

Приклад 1: Задати нечітке відношення «схожий менталітет» для таких національностей: українці (У), чехи (Ч), австрійці (А), німці (Н).

Використання звичайного (не нечіткого) відношення дозволяє виділити тільки одну пару націй зі схожими менталітетами – німців і австрійців. Таке відношення не враховує, що німецький менталітет ближчий до чеського, чим до українського. Нечітке відношення дозволяє легко представити ці знання. Один із варіантів такого відношення заданий наступною матрицею:

$$
\begin{array}{cc}
\begin{array}{cccc} У & Ч & А & Н \end{array} & \\
R = \begin{bmatrix} 1 & 0{,}5 & 0{,}2 & 0{,}1 \\ 0{,}5 & 1 & 0{,}6 & 0{,}3 \\ 0{,}2 & 0{,}6 & 1 & 0{,}8 \\ 0{,}1 & 0{,}3 & 0{,}8 & 1 \end{bmatrix} & \begin{array}{c} У \\ Ч \\ А \\ Н \end{array}
\end{array}
$$

Приклад 2: Задати нечітке відношення *x»y* («*x* приблизно дорівнює *y*»).

Нехай *x*Î{0,1,2,3} і *y*Î{0,1,2,3}. Тоді нечітке відношення можна задати матрицею вигляду:

$$
\begin{array}{cc}
\begin{array}{cccc} 0 & 1 & 2 & 3 \leftarrow y \end{array} & x \downarrow \\
R = \begin{bmatrix} 1 & 0{,}5 & 0{,}2 & 0{,}1 \\ 0{,}5 & 1 & 0{,}6 & 0{,}3 \\ 0{,}2 & 0{,}6 & 1 & 0{,}8 \\ 0{,}1 & 0{,}3 & 0{,}8 & 1 \end{bmatrix} & \begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \end{array}
\end{array}
$$

Для неперервних множин *X*=[0,3] і *Y*= [0,3] це нечітке відношення можна задати наступною функцією приналежності:

$$\mu_R(x,y) = e^{-0{,}2(x-y)^2}$$

Нечіткі відношення *x»y* на дискретних і неперервних носіях представлені на рис.1.

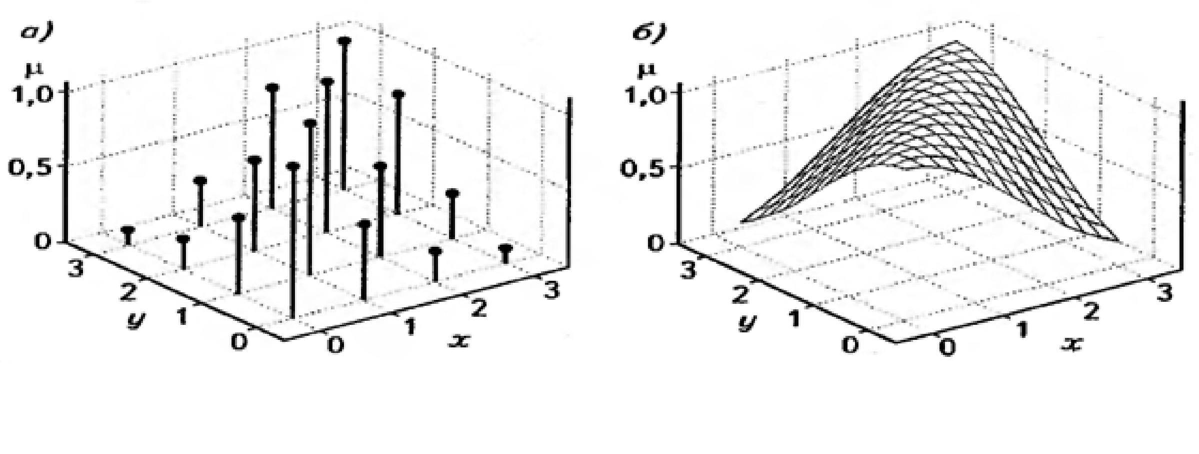


Рис.1 – Нечітке відношення x»y («x приблизно дорівнює y») а – відношення на дискретних множинах;

б – відношення на неперервних множинах

Приклад 3: Задати нечітке відношення $x<<y$ («x набагато менше, чим y»).

Нехай xÎ{0,1,2,3} і yÎ{0,1,2,3}. Тоді нечітке відношення можна задати матрицею вигляду:

$$R=\begin{array}{c} \begin{array}{cccc} 0 & 1 & 2 & 3\leftarrow y \end{array} \\ \begin{bmatrix} 0 & 0{,}2 & 0{,}6 & 1 \\ 0 & 0 & 0{,}2 & 0{,}6 \\ 0 & 0 & 0 & 0{,}2 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{array} \begin{array}{c} x\downarrow \\ 0 \\ 1 \\ 2 \\ 3 \end{array}$$

Для неперервних множин X=[0,3] і Y= [0,3] це нечітке відношення можна задати наступною функцією приналежності:

$$\mu_R(x,y)=\begin{cases} 0, & \text{если } x\geq y \\ \dfrac{1}{1+5/(x-y)^4}, & \text{если } x<y. \end{cases}$$

Нечіткі відношення $x<<y$ на дискретних і неперервних носіях представлені на рис.3.2.

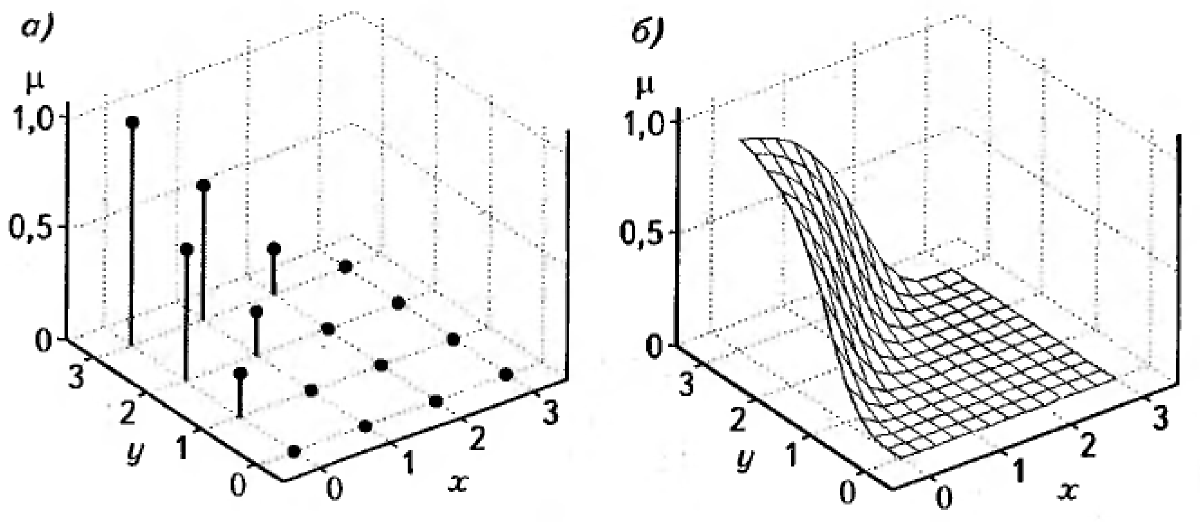


Рис.2 – Нечітке відношення *x*<<*y* («*x* набагато менше, чим *y*»)

а – відношення на дискретних множинах;

б – відношення на неперервних множинах

Нечіткі відношення дозволяють не лише зафіксувати сам факт відношення, але і вказати міру його виконання, що є важливим для багатьох практичних завдань.

Характеристики нечітких відношень

***Характеристиками нечітких відношень є*:** носій нечіткого відношення, відношення *α*-рівня, висота нечіткого відношення, нормальність, субнормальність, ядро, найближче чітке відношення, межі, точки переходу, опуклість, що визначаються подібно до нечітких множин: при цьому замість *x* використовують кортеж <x_1, x_2, …, x_n>, замість *A* використовують відношення *R*, а замість *E* – декартовий добуток E_1´E_2´…´E_n.

Носієм нечіткого відношення *R* на множинах *X* і *У* називається підмножина декартового добутку *X*´*Y* вигляду:

supp*R*={(*x*,*y*): (*x*,*y*)Î*X*´*Y*, μ_R(*x*, *y*)>0}.

Носій нечіткого відношення можна розглядати як звичайне відношення, що зв’язує всі пари (*x*, *y*)Î*X*´*Y*, для яких міра виконання нечіткого відношення *R* не дорівнює нулю. Кориснішим є використання *a*-перетинів нечіткого відношення, визначення яких аналогічні визначенням множин *a*-рівня.

a-перетином нечіткого відношення *R* на *X*´*Y* називається звичайне відношення, що зв’язує всі пари (*x*, *y*)Î*X*´*Y*, для яких міра виконання нечіткого відношення *R* не менше *a*:

R_a={(*x*,*y*): (*x*,*y*)Î*X*´*Y*, μ_R(*x*, *y*)³*a*}.

Модою нечіткого відношення *R* є кортеж w_m Î E_1´E_2´…´E_n, якщо цей кортеж є точкою локального максимуму відповідної функції приналежності μ_R(<x_1, x_2, …, x_n>), тобто виконується умова: w_m = arg max{μ_R(<x_1, x_2, …, x_n>)}.

Скінченим є нечітке відношення, якщо його носій є скінченим відношенням.

Рефлексивним є нечітке відношення R на $X \times X$, якщо для будь-якого $x \in X$ виконується рівність $\mu_R(x,x)=1$. У випадку кінцевої множини X всі елементи головної діагоналі матриці R дорівнюють 1. Прикладом рефлексивного нечіткого відношення може бути відношення «приблизно дорівнює».

Антирефлексивним є нечітке відношення R на $X \times X$, якщо для будь-якого $x \in X$ виконується рівність $\mu_R(x,x)=0$. У випадку кінцевої множини X всі елементи головної діагоналі матриці R дорівнюють 0. Прикладом рефлексивного нечіткого відношення може бути відношення «значно більше».

Симетричним є нечітке відношення R на $X \times Y$, якщо для будь-якої пари $(x, y) \in X \times Y$ виконується рівність $\mu_R(x,y) = \mu_R(y,x)$. Матриця симетричного нечіткого відношення, заданого на кінцевій множині, є симетричною.

Асиметричним є нечітке відношення R на $X \times Y$, якщо для будь-якої пари $(x, y) \in X \times Y$ справедливий вираз: $\mu_R(x,y) > 0 \Rightarrow \mu_R(y,x) = 0$. Прикладом асиметричного нечіткого відношення може бути відношення «набагато більше».

Зворотними є нечіткі відношення R та R^{-1} на $X \times Y$, якщо для будь-якої пари $(x, y) \in X \times Y$ виконується рівність:

$$\mu_R(x,y) = \mu_{R^{-1}}(y,x)$$

. Прикладом зворотних нечітких відношень може бути пара «набагато більше» – «набагато менше».

Транзитивним є нечітке відношення R на $X \times Y$, якщо $R \circ R \subseteq R$. Іншими словами, для будь-якої пари $(x, y) \in X \times Y$ ступінь виконання відношення R повинна бути не менше за ступінь виконання відношення $R \circ R$.

Транзитивним замиканням нечіткого відношення R є відношення:

$$\hat{R} = R \cup R^2 \cup R^3 \cup \ldots \cup R^n \cup \ldots$$

$$\hat{R} = \bigcup_{n=1,2,\ldots} R^n \text{, де } R^n = \underbrace{R \circ R \circ \ldots \circ R}_{n \text{ разів}} .$$

ТЕМА 4 МОДЕЛЮВАННЯ В MATLAB

FIS-структура Операції з нечіткою логікою у пакеті MATLAB дозволяє виконувати модуль Fuzzy Logic Toolbox. Він дозволяє створювати системи нечіткого логічного виведення і нечіткої класифікації в рамках середовища MatLab...

FIS-редактор ***FIS-редактор*** (рис.5.3) призначений для створення, збереження, завантаження і виведення у друк систем нечіткого логічного виведення, а також для редагування наступних властивостей...

Редактор функцій приналежності *Редактор функцій приналежності* (Membership Function Editor) призначений для завдання наступної інформації про терм-множини вхідних і вихідних...

Редактор бази знань ***Редактор бази знань*** (Rule Editor) призначений для формування і модифікації нечітких правил. Редактор бази знань може бути викликаний з будь-якого GUI-модуля, використовуваного із системами нечіткого логічного виведення...

Візуалізація нечіткого логічного виведення Візуалізація нечіткого логічного виведення здійснюється за допомогою GUI-модуля **Rule Viewer**. Цей модуль дозволяє проілюструвати хід логічного виведення за кожним правилом, одержання результуючої нечіткої множини і виконання процедури дефазифікації...

ANFIS-редактор

ANFIS є абревіатурою Adaptive Neuro-Fuzzy Inference System – адаптивна нейро-нечітка система. ANFIS-редактор дозволяє автоматично синтезувати з експериментальних даних нейро-нечіткі мережі...

Функції Fuzzy Logic Toolbox

*Функції, що входять до модуля Fuzzy Logic Toolbox можуть бути викликані з командного рядка. Для отримання переліку функцій слід ввести команду**: help fuzzy**. Наведемо короткий огляд функцій модуля Fuzzy Logic...*

FIS-редактор

FIS-редактор (рис.5.3) призначений для створення, збереження, завантаження і виведення у друк систем нечіткого логічного виведення, а також для редагування наступних властивостей: тип системи; найменування системи; кількість вхідних і вихідних змінних; найменування вхідних і вихідних змінних; параметри нечіткого логічного виведення.

Завантаження FIS-редактора відбувається за допомогою команди **fuzzy**. У результаті з'являється інтерактивне графічне вікно. У нижній частині графічного вікна FIS-редактори розташовані кнопки **Help** і **Close**, що дозволяють викликати вікно довідки і закрити редактор, відповідно.

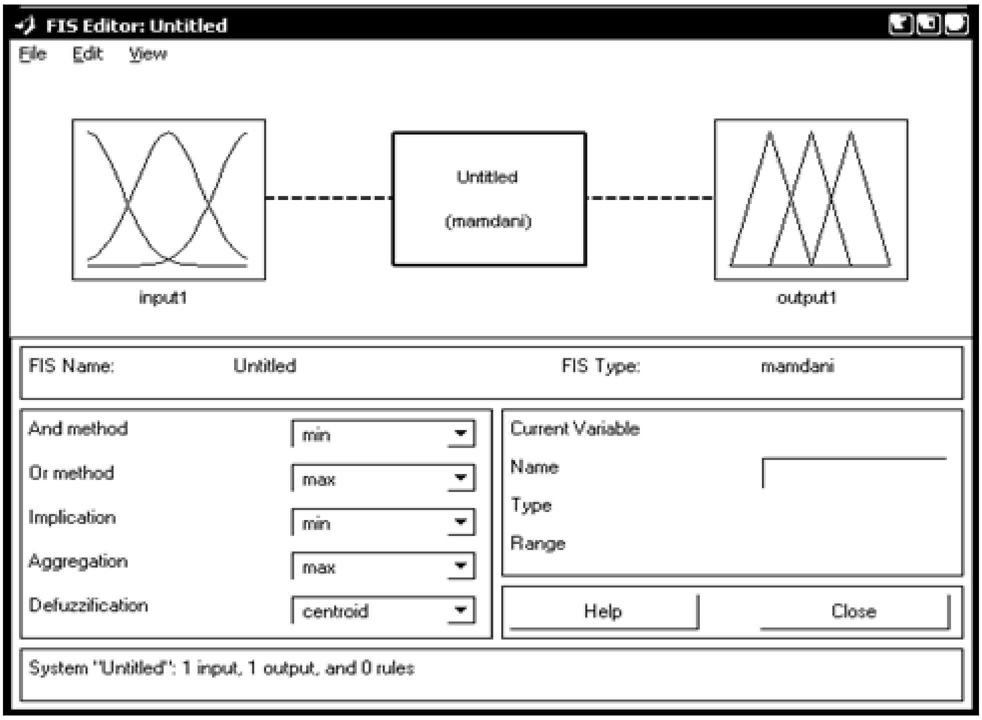
FIS-редактор містить 8 меню. Це три загальносистемних меню – **File**, **Edit**, **View**, і п'ять меню для вибору параметрів нечіткого логічного виведення –

And Method, **Or Method**, **Implication**, **Aggregation** і **Defuzzification**.

Меню File – це загальне меню для всіх GUI-модулів, використовуваних із системами нечіткого логічного виведення.

За допомогою команди **New FIS...** користувач має можливість створити нову систему нечіткого логічного виведення. При виборі цієї команди з'являться дві альтернативи: **Mamdani** і **Sugeno**, що визначають тип створюваної системи. Створити систему типу Mamdani можна також натисканням Ctrl+N.

За допомогою команди Import користувач має можливість завантажити раніше створену систему нечіткого логічного виведення. При виборі цієї команди з'являться дві альтернативи **From Workspace...** і **From disk**, що дозволяють завантажити систему нечіткого логічного виведення з робочої області MatLab і з диска, відповідно. При виборі команди **From Workspace...** з'явиться діалогове вікно, у якому необхідно вказати ідентифікатор системи нечіткого логічного виведення, що знаходиться в робочій області MatLab. При виборі команди **From disk** з'явиться діалогове вікно, у якому необхідно вказати ім'я файлу системи нечіткого логічного виведення. Файли систем нечіткого логічного виведення мають розширення .fis . Завантажити систему нечіткого логічного виведення з диска можна також натисканням Ctrl+N чи командою fuzzy FIS_name, де FIS_name – ім'я файлу системи нечіткого логічного виведення.



При виборі команди **Export** з'являться дві альтернативи **To Workspace...** і **To disk**, що дозволяють скопіювати систему нечіткого логічного виведення в робочу область MatLab і на диск, відповідно. При виборі команди **To Workspace...** з'явиться діалогове вікно, у якому необхідно вказати ідентифікатор системи нечіткого логічного виведення, під яким вона буде збережена в робочій області MatLab. При виборі команди **To disk** з'явиться діалогове вікно, у якому необхідно вказати ім'я файлу системи нечіткого логічного виведення. Скопіювати систему нечіткого логічного виведення в робочу область і на диск можна також натисканням Ctrl+T і Ctrl+S, відповідно.

Команда **Print** дозволяє вивести на принтер копію графічного вікна. Печатка можлива також по натисканню Ctrl+P.

Команда **Close** закриває графічне вікно. Закриття графічного вікна відбувається по натисканню Ctrl+W чи однократному клику лівою кнопкою миші по кнопці Close.

Меню Edit:

Команда **Undo** скасовує раніше зроблену дію. Виконується також по натисканню Ctrl+Z.

Команда **Add Variable...** дозволяє додати в систему нечіткого логічного виведення ще одну змінну. При виборі цієї команди з'являться дві альтернативи **Input** і **Output**, що дозволяють додати вхідну і вихідну змінну, відповідно.

Команда **Remove Selected Variable** видаляє поточну змінну із системи. Ознакою поточної змінної є червона окантовка її прямокутника. Призначення поточної змінної відбувається за допомогою однократного клика лівої кнопки миші по її прямокутнику. Видалити поточну змінну можна також за допомогою натискання Ctrl+X.

Команда **Membership Function...** відкриває редактор функцій приналежностей. Ця команда може бути також виконана натисканням Ctrl+2.

Команда **Rules…** відкриває редактор бази знань. Ця команда може бути також виконана натисканням Ctrl+3.

Меню View – це загальне меню для всіх GUI-модулів, використовуваних із системами нечіткого логічного виведення. Дане меню дозволяє відкрити вікно візуалізації нечіткого логічного виведення (команда Rules або натискання клавіш Ctrl+5) і вікно виведення поверхні "вхід-вихід", що відповідає системі нечіткого логічного виведення (команда Surface або натискання клавіш Ctrl+6).

Меню And Method – це меню дозволяє установити наступні реалізації логічної операції "ТА": min – мінімум; prod – множення.

Користувач також має можливість установити власну реалізацію операції "ТА". Для цього необхідно вибрати команду **Custom…** і в графічному вікні, що з'явилося, надрукувати ім'я функції, що реалізує цю операцію.

Меню Or Method – це меню дозволяє установити наступні реалізації логічної операції "АБО": **max** – множення; **probor** - імовірнісне "АБО".

Користувач також має можливість установити власну реалізацію операції "АБО". Для цього необхідно вибрати команду **Custom…** і в графічному вікні, що з’явилося, надрукувати ім’я функції, що реалізує цю операцію.

Меню Implication – це меню дозволяє установити наступні реалізації імплікації: **min** – мінімум; **prod** – множення.

Користувач також має можливість установити власну реалізацію імплікації. Для цього необхідно вибрати команду **Custom…** і в графічному вікні, що з’явилося, надрукувати ім’я функції, що реалізує цю операцію.

Меню Aggregation – це меню дозволяє установити наступні реалізації операції об’єднання функцій приналежності вихідної змінної: **max** – максимум; **sum** – сума; **probor** – імовірнісне "АБО".

Користувач також має можливість установити власну реалізацію цієї операції. Для цього необхідно вибрати команду **Custom…** і в графічному вікні, що з’явилося, надрукувати ім’я функції, що реалізує цю операцію

Меню Defuzzification – це меню дозволяє вибрати метод дефазифікації. Для систем типу Мамдані запрограмовані наступні методи: **centroid** – центр ваги; **bisector** – медіана; **lom** – найбільший з максимумів; **som** – найменший з максимумів; **mom** – середнє з максимумів. Для систем типу Сугено запрограмовані наступні методи: **wtaver** – зважене середнє; **wtsum** – зважена сума.

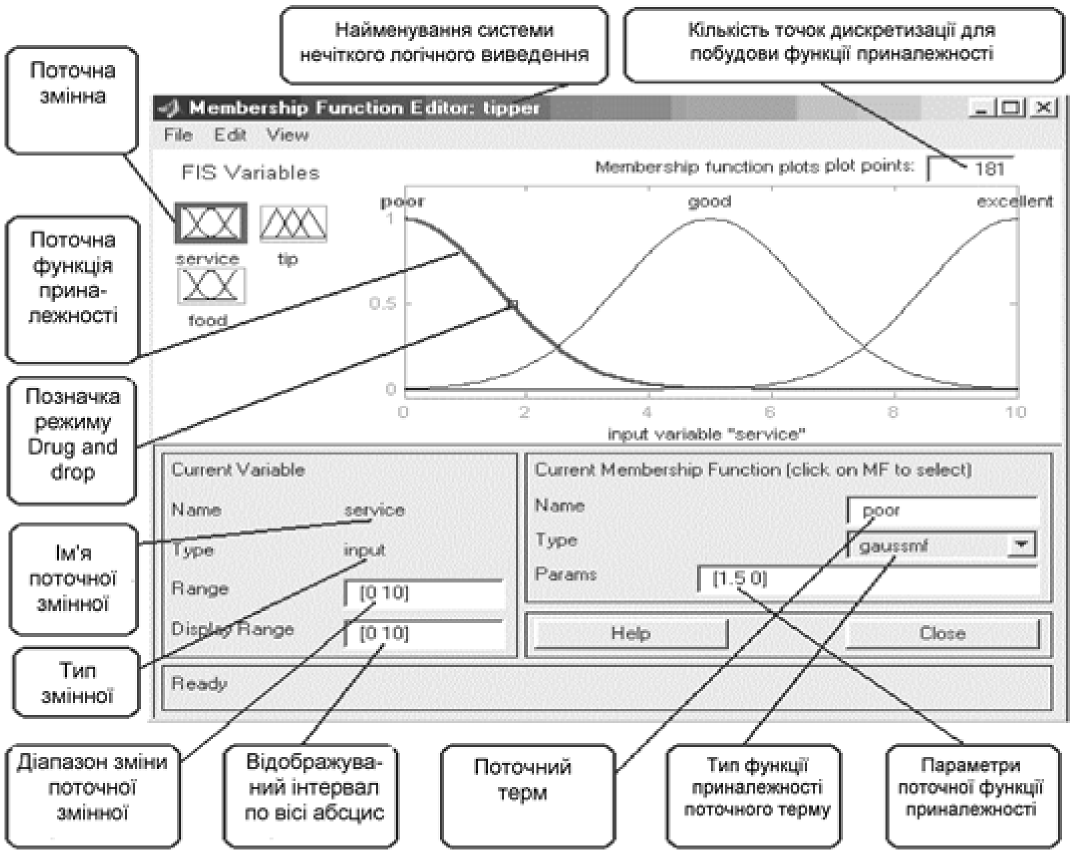
Користувач також має можливість установити власний метод дефазифікації. Для цього необхідно вибрати команду **Custom…** і в графічному вікні, що з’явилося, надрукувати ім’я функції, що реалізує цю операцію.

Редактор функцій приналежності

Редактор функцій приналежності (Membership Function Editor) призначений для завдання наступної інформації про терм-множини вхідних і вихідних змінних: кількість термів; найменування термів; тип і параметри функцій приналежності, що необхідні для представлення лінгвістичних термів у вигляді нечітких множин.

Редактор функцій приналежності може бути викликаний з будь-якого GUI-модуля, використовуваного із системами нечіткого логічного виведення, командою Membership Functions... меню Edit або натисканням клавіш Ctrl+2.

У FIS-редакторі відкрити редактор функцій приналежності можна також подвійним кликом лівою кнопкою миші по полю вхідної або вихідної змінних. Загальний вид редактора функцій приналежності з указівкою функціонального призначення основних полів графічного вікна приведений на рисунку.



Основний опис

У нижній частині графічного вікна розташовані кнопки **Help** і **Close**, що дозволяють викликати вікно довідки і закрити редактор, відповідно.

Редактор функцій приналежності містить чотири меню – **File**, **Edit**, **View**, **Type** і чотири вікна введення інформації – **Range**, **Display Range**, **Name** і **Params**. Ці чотири вікна призначені для завдання діапазону зміни поточної змінної, діапазону виведення функцій приналежності, найменування поточного лінгвістичного терму і параметрів його функції приналежності, відповідно.

Параметри функції приналежності можна підбирати й у графічному режимі, шляхом зміни форми функції приналежності за допомогою технології "Drug and drop". Для цього необхідно позиціонувати курсор миші на знаку режиму "Drug and drop", натиснути на ліву кнопку миші і не відпускаючи її змінювати форму функції приналежності. Параметри функції приналежності будуть перераховуватися автоматично.

Меню Edit:

Команда Undo скасовує раніше зроблену дію. Виконується також по натисканню Ctrl+Z.

Команда **Add MFs…** дозволяє додати терми в терм-множину, використовувану для лінгвістичної оцінки поточної змінної. При виборі цієї команди з'явиться діалогове вікно, у якому необхідно вибрати тип функції приналежності і кількість термів. Значення параметрів функцій приналежності будуть встановлені автоматично таким чином, щоб рівномірно покрити область визначення змінної, заданої у вікні **Range**. При зміні області визначення у вікні Range параметри функцій приналежності будуть промасштабовані.

Команда **Add Custom MF…** дозволяє додати одних лінгвістичний терм, функція приналежності якого відрізняється від убудованих. Після вибору цієї команди з'явиться графічне вікно, у якому необхідно надрукувати лінгвістичний терм (поле **MF name**), ім'я функції приналежності (поле **M-File function name**) і параметри функції приналежності (поле **Parameter list**).

Команда **Remove Selected MF** видаляє поточний терм із терм-множини поточної змінної. Ознакою поточної змінної є червона окантовка її прямокутника. Ознакою поточного терму є червоний колір його функції приналежності. Для вибору поточного терму необхідно провести позиціонування курсору миші на графіку функції приналежності і зробити клик лівою кнопкою миші.

Команда **Remove All MFs** видаляє всі терми з терм-множини поточної змінної.

Команда **FIS Properties…** відкриває FIS-редактор. Ця команда може бути також виконана натисканням Ctrl+1.

Команда **Rules…** відкриває редактор бази знань. Ця команда може бути також виконана натисканням Ctrl+3.

Меню Type дозволяє установити тип функцій приналежності термів, використовуваних для лінгвістичної оцінки поточної змінної.

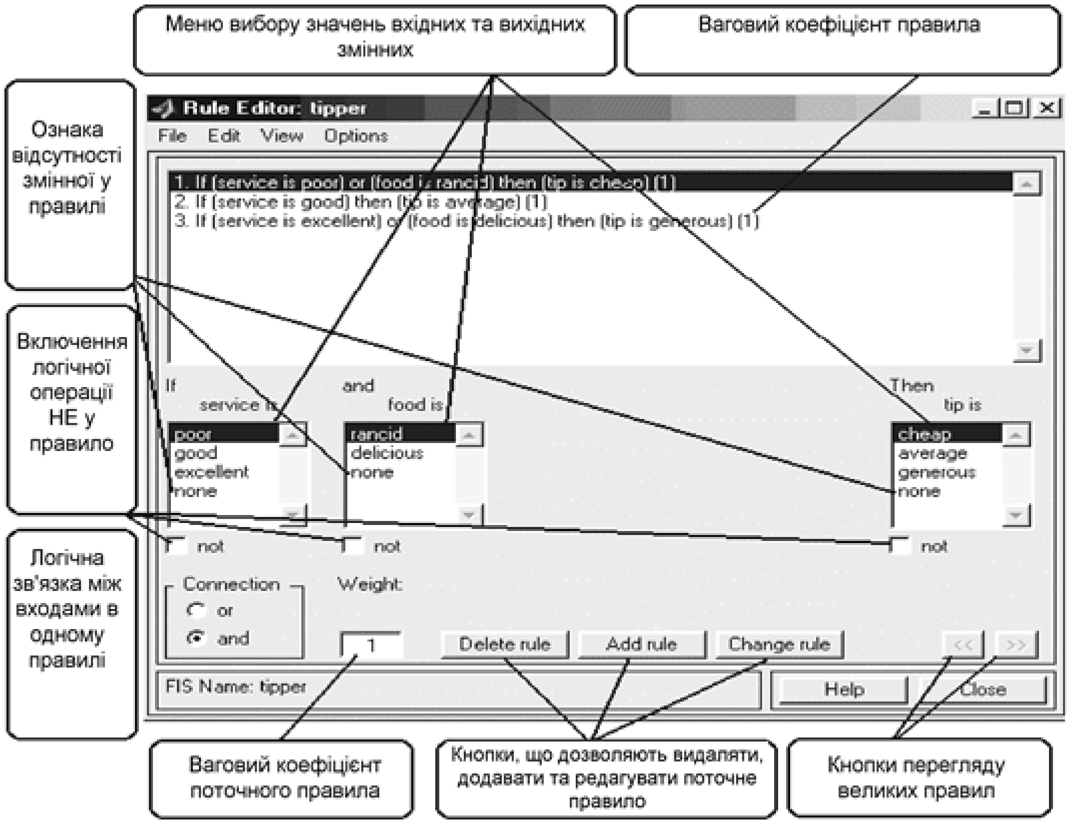
Редактор бази знань

Редактор бази знань (Rule Editor) призначений для формування і модифікації нечітких правил. Редактор бази знань може бути викликаний з будь-якого GUI-модуля, використовуваного із системами нечіткого логічного виведення, командою **Rules…** меню **Edit** або натисканням клавіш Ctrl+3. У FIS-редакторі відкрити редактор бази знань можна також подвійним кликом лівою кнопкою миші по прямокутнику з назвою системи нечіткого логічного виведення, розташованого в центрі графічного вікна.

Загальний вид редактора бази знань із указівкою функціонального призначення основних полів графічного вікна приведений на рисунку.

У нижній частині графічного вікна розташовані кнопки **Help** і **Close**, що дозволяють викликати вікно довідки і закрити редактор, відповідно.

Редактор функцій приналежності містить чотири системних меню **File**, **Edit**, **View**, **Options**, меню вибору термів вхідних і вихідних змінних, поля установки логічних операцій ТА, АБО, НЕ і ваг правил, а також кнопки редагування і перегляду правил.



Для введення нового правила в базу знань необхідно за допомогою миші вибрати відповідну комбінацію лінгвістичних термів вхідних і вихідних змінних, установити тип логічного зв'язування (ТА або АБО) між змінними усередині правила, установити наявність чи відсутність логічної операції НЕ для кожної лінгвістичної змінної, увести значення вагового коефіцієнта правила і натиснути кнопку **Add Rule**. За замовчуванням установлені наступні параметри: логічне зв'язування змінних усередині правила – ТА; логічна операція НЕ – відсутня; значення вагового коефіцієнта правила – 1.

Можливі випадки, коли істинність правила не змінюється при довільному значенні деякої вхідної змінної, тобто ця змінна не впливає на результат нечіткого логічного виведення в даній області факторного простору. Тоді лінгвістичне значення цієї змінної необхідно установити як none.

Для видалення правила з бази знань необхідно зробити однократний клик лівою кнопкою миші на цьому правилі та натиснути кнопку **Delete Rule**.

Для модифікації правила необхідно зробити однократний клик лівою кнопкою миші на цьому правилі, потім установити необхідні параметри правила і натиснути кнопку **Edit Rule**.

Меню редактора бази знань MatLab

Меню Edit:

Команда **Undo** скасовує раніше зроблену дію. Виконується також по натисканню Ctrl+Z.

Команда **FIS Properties…** відкриває FIS-редактор. Ця команда може бути також виконана натисканням Ctrl+1.

Команда **Membership Function…** відкриває редактор функцій приналежностей. Ця команда може бути також виконана натисканням Ctrl+2.

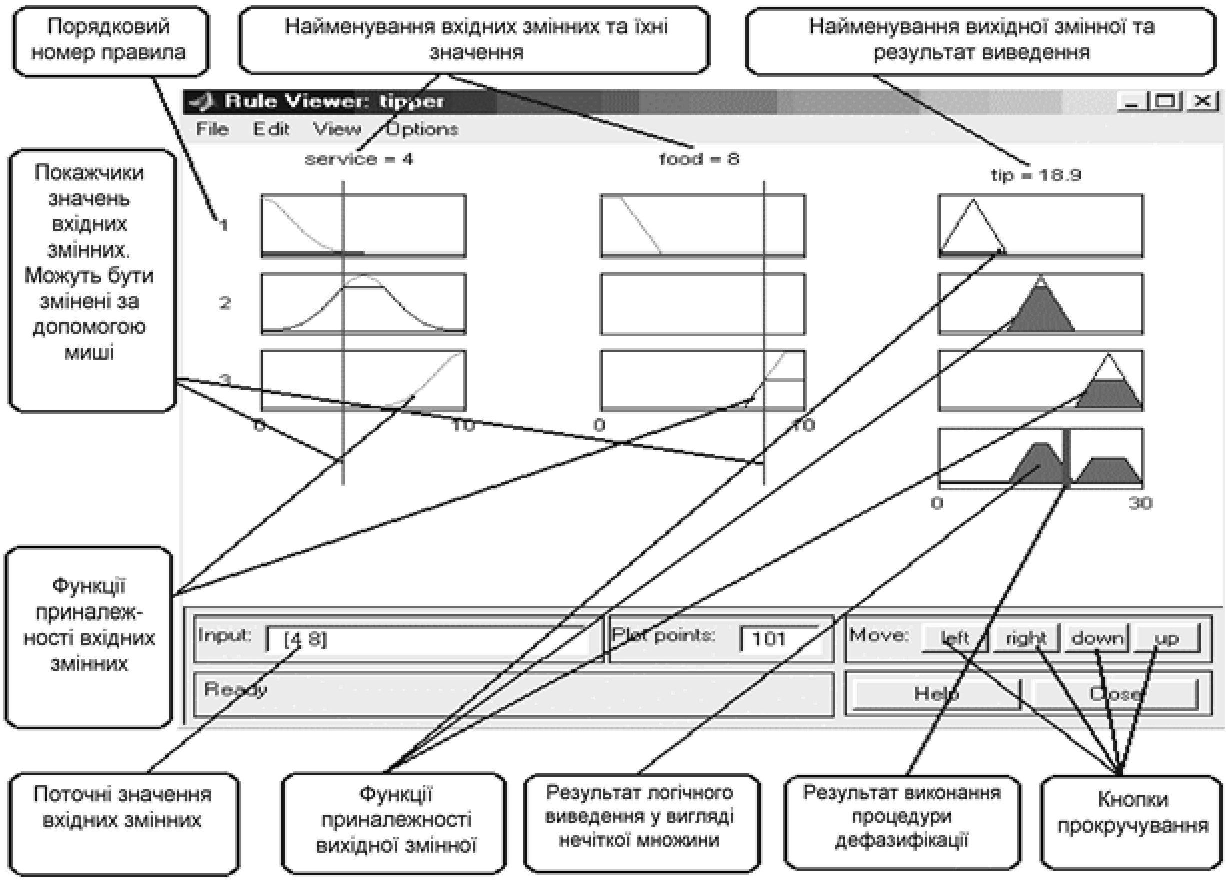
Меню Options дозволяє установити мову і формат правил бази знань. При виборі команди Language з'явиться список мов English (Англійська), Deutsch (Німецька), Francais (Французька), з якого необхідно вибрати одну.

При виборі команди **Format** з'явиться список можливих форматів правил бази знань: **Verbose** – лінгвістичний; **Symbolic** – логічний; **Indexed** – індексований.

Візуалізація нечіткого логічного виведення

Візуалізація нечіткого логічного виведення здійснюється за допомогою GUI-модуля **Rule Viewer**. Цей модуль дозволяє проілюструвати хід логічного виведення за кожним правилом, одержання результуючої нечіткої множини і виконання процедури дефазифікації. **Rule Viewer** може бути викликаний з будь-якого GUI-модуля, використовуваного із системами нечіткого логічного виведення, командою **View rules …** меню **View** чи натисканням клавіш Ctrl+4. Вид **Rule Viewer** для системи логічного виведення tipper із указівкою функціонального призначення основних полів графічного вікна приведений на рисунку.

Rule Viewer містить чотири меню – **File**, **Edit**, **View**, **Options**, два поля уведення інформації – **Input** і **Plot points** та кнопки прокручування зображення вліво – вправо (left-right), униз (up-down).



У нижній частині графічного вікна розташовані також кнопки **Help** і **Close**, що дозволяють викликати вікно довідки і закрити редактор, відповідно. Кожне правило бази знань представляється у виді послідовності горизонтально розташованих прямокутників. При цьому перші два прямокутники відображають функції приналежностей термів посилки правила (Якщо – частина правила), а останній третій прямокутник відповідає функції приналежності терму-наслідку вихідної змінної (То – частина правила).

Порожній прямокутник у візуалізації другого правила означає, що в цьому правилі посилка за змінною food відсутня (food is none). Жовте заливання графіків функцій приналежностей вхідних змінних указує наскільки значення входів, відповідають термам даного правила. Для виведення правила у форматі **Rule Editor** необхідно зробити однократний клик лівої кнопки миші по номеру відповідного правила. У цьому випадку зазначене правило буде виведено в нижній частині графічного вікна.

Блакитне заливання графіка функції приналежності вихідної змінної являє собою результат логічного виведення у вигляді нечіткої множини за даним правилом. Результуючу нечітку множину, що відповідає логічному виведенню за всіма правилами, показано в нижньому прямокутнику останнього стовпця графічного вікна. У цьому ж прямокутнику червона вертикальна лінія відповідає чіткому значенню логічного виведення, отриманого в результаті дефазифікації.

Уведення значень вхідних змінних може здійснюватися двома способами: шляхом уведення чисельних значень у поле **Input**; за допомогою миші, шляхом переміщення ліній-покажчиків червоного кольору.

В останньому випадку необхідно позиціонувати курсор миші на червоній вертикальній лінії, натиснути на ліву кнопку миші і не відпускаючи неї перемістити покажчик на потрібну позицію. Нове чисельне значення відповідної вхідної змінної буде перелічено автоматично і виведене у вікно **Input**.

У поле **Plot points** задається кількість точок дискретизації для побудови графіків функцій приналежності. Значення за замовчуванням – 101.

Меню системи tipper

Меню Edit:

Команда **FIS Properties…** відкриває FIS-редактор. Ця команда може бути також виконана натисканням Ctrl+1.

Команда **Membership Functions…** відкриває редактор функцій приналежностей. Ця команда може бути також виконана натисканням Ctrl+2.

Команда **Rules…** відкриває редактор бази знань. Ця команда може бути також виконана натисканням Ctrl+3.

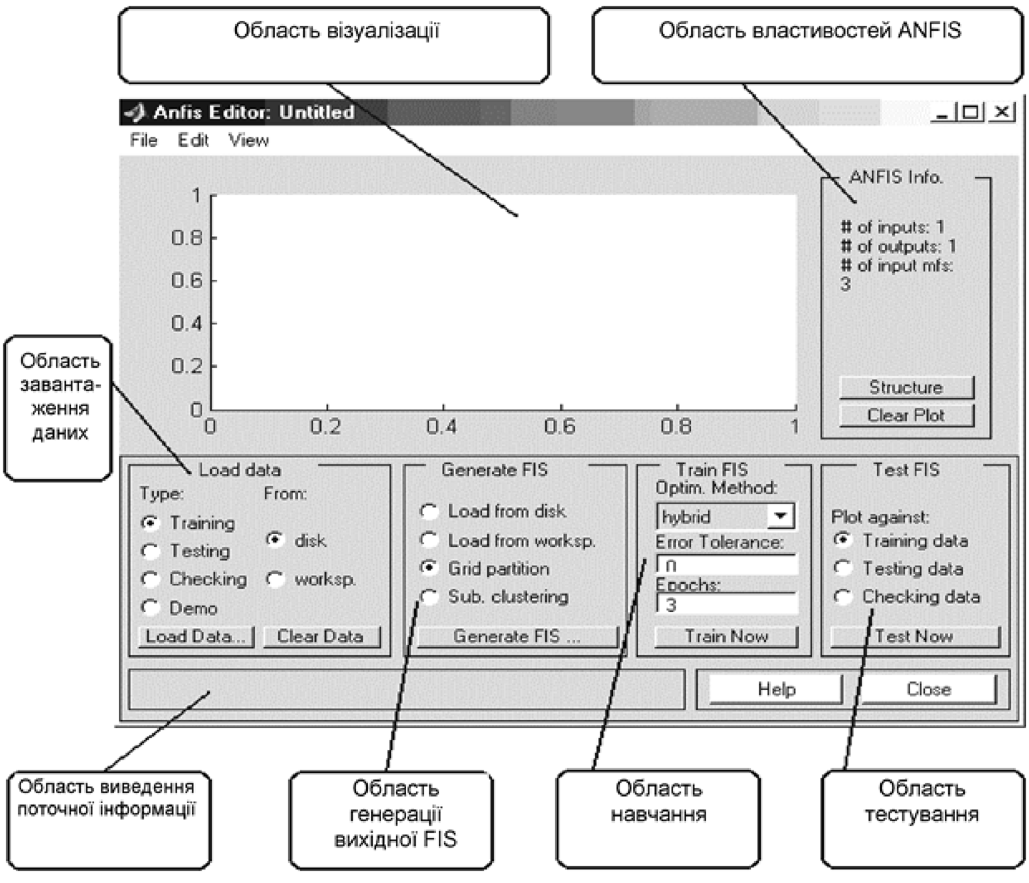
Меню Options містить тільки одну команду **Format**, що дозволяє установити один з наступних форматів виведення обраного правило в нижній частині графічного вікна: **Verbose** – лінгвістичний; **Symbolic** – логічний;

Indexed – індексований.

ANFIS-редактор

ANFIS є абревіатурою Adaptive Neuro-Fuzzy Inference System – адаптивна нейро-нечітка система. ANFIS-редактор дозволяє автоматично синтезувати з експериментальних даних нейро-нечіткі мережі. Нейро-нечітку мережу можна розглядати як один з різновидів систем нечіткого логічного виведення типу Суджено. При цьому функції приналежності синтезованих систем налагоджено (навчено) так, щоб мінімізувати відхилення між результатами нечіткого моделювання й експериментальних даних.

Завантаження **ANFIS-редактора** здійснюється за командою **anfisedit**. У результаті виконання цієї команди з'явиться графічне вікно, зображене на рисунку. На цьому ж рисунку зазначено функціональні області ANFIS-редактора, опис яких приведено нижче.



ANFIS-редактор містить 3 верхніх меню – **File**, **Edit** і **View**, область візуалізації, область властивостей ANFIS, область завантаження даних, область генерування вихідної системи нечіткого логічного виведення, область навчання, область тестування, область виведення поточної інформації, а також кнопки **Help** і **Close**, що дозволяють викликати вікно довідки і закрити ANFIS-редактор, відповідно.

Меню та команди Anfis редактора

Меню Edit:

Команда **Undo** скасовує раніше зроблену дію. Виконується також по натисканню Ctrl+Z.

Команда **FIS Properties…** відкриває FIS-редактор. Ця команда може бути також виконана натисканням Ctrl+1.

Команда **Membership Functions…** відкриває редактор функцій приналежності. Ця команда може бути також виконана натисканням Ctrl+2.

Команда **Rules…** відкриває редактор бази знань. Ця команда може бути також виконана натисканням Ctrl+3.

Команда **Anfis…** відкриває ANFIS-редактор. Ця команда може бути також виконана натисканням Ctrl+3. Відмітимо, що дана команда, запущена з ANFIS-редактора не приводить до виконання яких-небудь дій, бо цей редактор уже відкритий. Однак, у меню **Edit** інших GUI-модулів, використовуваних із системами нечіткого логічного виведення, додається команда **Anfis…** , що дозволяє відкрити ANFIS-редактор з цих модулів.

Область візуалізації містить два типи інформації: *при навчанні системи* – крива навчання у виді графіка залежності помилки навчання від порядкового номера ітерації; *при завантаженні даних і тестуванні системи* – експериментальні дані і результати моделювання.

Експериментальні дані і результати моделювання виводяться у виді множини точок у двовимірному просторі. При цьому по вісі абцис відкладається порядковий номер рядка даних у вибірці (навчальній, тестовій або контрольній), а по осі ординат – значення вихідної змінної для даного рядка вибірки. Використовуються наступні маркери: блакитна точка (**.**) – тестова вибірка; блакитна окружність (o) – навчальна вибірка; блакитний плюс (+) – контрольна вибірка; червона зірочка (*) – результати моделювання.

В ***області властивостей*** ANFIS (**ANFIS info**) виводиться інформація про кількість вхідних і вихідних змінних, про кількість функцій приналежності для кожної вхідної змінної, а також про кількість рядків у вибірках. У цій області розташовані дві кнопки **Structure** і **Clear Plot**.

Натискання кнопки Structure відчиняє нове графічне вікно, у якому система нечіткого логічного виведення представлена у виді нейро-нечіткої мережі. Натискання кнопки **Clear Plot** дозволяє очистити область візуалізації.

В ***області завантаження даних*** (**Load data**) розташовані: меню вибору типу даних (**Type**), що містить альтернативи (**Traning** – навчальна вибірка; **Testing** – тестова вибірка; **Checking** – контрольна вибірка; **Demo** – демонстраційний приклад); меню вибору джерела даних (**From**), що містить альтернативи (**disk** – диск; **worksp** – робоча область MatLab); кнопка завантаження даних **Load Data…**, по натисканню якої з'являється діалогове вікно вибору файлу, якщо завантаження даних відбувається з диска, або вікно введення ідентифікатора вибірки, якщо завантаження даних походить з робочої області; кнопка очищення даних **Clear Data**.

В ***області генерування*** (**Generate FIS**) розташовані меню вибору способу створення вихідної системи нечіткого логічного виведення. Меню містить наступні альтернативи: **Load from disk** – завантаження системи з диска; **Load from worksp.** – завантаження системи з робочої області MatLab; **Grid partition** – генерування системи по методу ґрат (без кластеризації); **Sub. clustering** – генерування системи за методом субкластеризації.

В області також розташована кнопка **Generate**, по натисканню якої генерується вихідна система нечіткого логічного виведення.

При виборі **Load from disk** з'являється стандартне діалогове вікно відкриття файлу.

При виборі **Load from worksp.** з'являється стандартне діалогове вікно введення ідентифікатора системи нечіткого логічного виведення.

При виборі **Grid partition** з'являється вікно введення параметрів методу ґрат, у якому потрібно вказати кількість термів для кожен вхідний змінної і тип функцій приналежності для вхідних і вихідної змінних.

При виборі **Sub. clustering** з'являється вікно введення наступних параметрів методу субкластеризації: **Range of influence** – рівні впливу вхідних змінних; **Squash factor** – коефіцієнт пригнічення; **Accept ratio** – коефіцієнт, що встановлює у скільки разів потенціал даної точки повинен бути вище потенціалу центра першого кластера для того, щоб центром одного з кластерів була призначена розглянута точка; **Reject ratio** – коефіцієнт, що встановлює у скільки разів потенціал даної точки повинен бути нижче потенціалу центра першого кластера, щоб розглянута точка була виключена з можливих центрів кластерів.

В ***області навчання*** (**Train FIS**) розташовані меню вибору методу оптимізації (**Optim. method**), поле завдання необхідної точності навчання (**Error tolerance**), поле завдання кількості ітерацій навчання (**Epochs**) і кнопка **Train Now**, натискання якого запускає режим навчання. Проміжні результати навчання виводяться в область візуалізації й у робочу область MatLab. У ANFIS-редакторі реалізовані два методи навчання: **backpropa** – метод зворотного поширення помилки, заснований на ідеях методу найшвидшого спуска; **hybrid** – гібридний метод, що поєднує метод зворотного поширення помилки з методом найменших квадратів.

В ***області тестування*** (**Test FIS**) розташовані меню вибору вибірки і кнопка **Test Now**, при натисканню на яку відбувається тестування нечіткої системи з виведення результатів в область візуалізації.

Область виведення поточної інформації: у цій області виводиться найбільш істотна поточна інформація, наприклад, повідомлення про закінчення виконання операцій, значення помилки чи навчання тестування і т.п.

Функції Fuzzy Logic Toolbox

Функції, що входять до модуля Fuzzy Logic Toolbox можуть бути викликані з командного рядка. Для отримання переліку функцій слід ввести команду**: help fuzzy**. Наведемо короткий огляд функцій модуля Fuzzy Logic.

Редактори з графічним інтерфейсом користувача: **anfisedit** – інструмент для навчання та тестування ANFIS; **findcluster** – інструмент для кластеризації; **fuzzy** – базовий редактор FIS; **mfedit** – редактор функцій приналежності; **ruleedit** – редактор та аналізатор правил; **ruleview** – демонстратор правил та діаграм нечіткого виведення; **surfview** – демонстратор вихідної поверхні.

Функції приналежності: dsigmf, gauss2mf, gaussmf, gbellmf, pimf, psigmf, smf, sigmf, trapmf, trimf, zmf.

Модуль Fuzzy Logic Toolbox пакету MATLAB включає 11 убудованих функцій приналежності, що використовують такі основні функції: кусочно-лінійну; гаусівській розподіл; сигмоїдну криву; квадратичну та кубічну криві.

Для зручності імена всіх убудованих функцій приналежності закінчуються на mf. Виклик функції приналежності здійснюється в такий спосіб: ***namemf(x, params)***, де ***namemf*** – найменування функції приналежності; ***x*** – вектор, для координат якого необхідно розрахувати значення функції приналежності; ***params*** – вектор параметрів функції приналежності.

Найпростіші функції приналежності ***трикутна*** (**trimf**) і ***трапецієподібна*** (**trapmf**) формуються з використанням кусочно-лінійної апроксимації. Трапецієподібна функція приналежності є узагальненням трикутної, вона дозволяє задавати ядро нечіткої множини у виді інтервалу. У випадку трапецієподібної функції приналежності можлива наступна зручна інтерпретація: ядро нечіткої множини – оптимістична оцінка; носій нечіткої множини – песимістична оцінка.

Дві функції приналежності – ***симетрична гаусівська*** (**gaussmf**) і ***двостороння гаусівська*** (**gaussmf**) формуються з використанням гаусівського розподілу. Функція **gaussmf** дозволяє задавати асиметричні функції приналежності. Узагальнена колоколообразна функція приналежності (**gbellmf**) за своєю формою схожа на гаусівські. Ці функції приналежності часто використовуються в нечітких системах, тому що на всій області визначення вони є гладкими і приймають ненульові значення.

Функції приналежності **sigmf**, **dsigmf**, **psigmf** засновані на використанні ***сигмоїдної кривої***. Ці функції дозволяють формувати функції приналежності, значення яких починаючи з деякого значення аргументу і до +(–) ∞ рівні 1. Такі функції зручні для завдання лінгвістичних термів типу «високий» або «низький».

Поліноміальна апроксимація застосовується при формуванні функцій **zmf**, **pimf** і **smf**, графічні зображення яких схожі на функції **sigmf**, **dsigmf**, **psigmf**, відповідно.

У Fuzzy Logic Toolbox передбачена можливість для користувача створення власної функції приналежності. Для цього необхідно створити m-функцію, що містить два вхідних аргументи – вектор, для координат якого необхідно розрахувати значення функції приналежності і вектор параметрів функції приналежності. Вихідним аргументом функції повинний бути вектор ступенів приналежності.

Функції FIS: **addmf** – додає функцію приналежності до FIS; **addrule** – додає правило до FIS; **addvar** – додає змінну до FIS; **defuzz** – дефазифікує функцію приналежності; **evalfis** – здійснює обчислення нечіткого виведення; **evalmf** – обчислює функцію приналежності; **gensurf** – генерує поверхню виходу FIS; **getfis** – повертає властивості нечіткої системи; **mf2mf** – транслює параметри між функціями прииналежності; **newfis** – створює нову FIS; **parsrule** – аналізує нечіткі правила; **plotfis** – показує діаграму "вхід-вихід" для FIS; **plotmf** – показує усі функції приналежності для однієї змінної; **readfis** – завантажує FIS з диску; **rmmf** – видаляє функцію приналежності з

FIS; **rmvar** – видаляє змінну з FIS; **setfis** – встановлює властивості нечіткої системи; **showfis** - показує анотовану FIS; **showrule** – відображує правила FIS; **writefis** – зберігає FIS на диску.

Функція **output=evalfis(input, fis, numofpoints)** виконує обчислення для вибірки екземплярів за допомогою вказаної нейро-нечіткої мережі. Результатом є обчислені виходи функції, яку апроксимує нейро-нечітка мережа. Аргументи та результат функції: **input** – входи обчислюваної вибірки; **fis** – нейро-нечітка мережа; **numofpoints** – кількість точок для проведення дефазифікації (рекомендується брати значення не менше 100, зменшення цього параметру прискорює процес обчислень, але зменшує точність); **output** – обчислені виходи.

Вибірки даних та нейро-нечіткі мережі, з якими працюють описані функції, повинні зберігатися в робочій області MATLAB (в оперативній пам'яті). В стовпцях матриць, що представляють собою вибірки, зберігаються значення входів (ознак) та виходів, в рядках – екземпляри вибірки.

Для завантаження вибірок з диску або запису на диск необхідно використовувати команду MATLAB: **s=load(filename)**, яка завантажує вміст файлу в змінну s. Функція **save(filename, s)** зберігає змінну s у файл.

Для можливості використання створеної нейро-нечіткої мережі в наступних сеансах роботи або використання мережі з попередніх слід використовувати такі функції модуля Fuzzy Logic Toolbox: **writefis(fis,filename)** – зберігає нейро-нечітку мережу fis в файл; **fis=readfis(filename)** – завантажує з файлу нейро-нечітку мережу в змінну fis.

Передові технології: **anfis** – функція навчання для системи Суджено; функції кластер-аналізу: **fcm**, **genfis1**, **genfis2**; **subclust**.

Різні функції: **convertfis** – перетворює нечітку матрицю структури версії 1.0 у матрицю структури версії 2.0; **discfis** – дискретизує FIS; **evalmmf** – використовується для обчислення множинних функцій приналежності; fstrvcat – поєднує матриці різного розміру; **fuzarith** – функція нечіткої арифметики; **findrow** – шукає рядки матриці, що відповідають вхідному рядку; **genparam** – генерує початкові параметри передумов для навчання ANFIS; **probor** – імовірнісне АБО; **sugmax** – максимальний вихідний діапазон для системи Суджено.

Функція **C = fuzarith(X, A, B, OPERATOR)** реалізує базові операції нечіткої логіки та повертає нечітку множину С як результат застосування оператора OPERATOR до нечітких множин А та В з універсальної множини Х. Змінні А, В та Х мають бути векторами однакової розмірності. OPERATOR має бути одним з рядків: **'sum'** – сума, **'sub'** – вирахування, **'prod'** – добуток, **'div'** – ділення. Нечітка множина С, яка повертається, є вектор-стовпцем тієї є довжини, що й А та В. Зауважимо, що ця функція використовує інтервальну арифметику та передбачає, що: А та В є опуклими нечіткими множинами; функції приналежності для А та В поза Х є нулем.

Допоміжні функції Fuzzy toolbox

***Допоміжні функції графічного користувальницького інтерфейсу*:** **cmfdlg** – додає діалог вибору функцій приналежності; **cmthdlg** – додає діалог вибору методу виведення; **fisgui** – дискрипторне посилання на інтерфейсні засоби модуля Fuzzy Logic Toolbox; **gfmfdlg** – генерує FIS з використанням діалогу методу ґрат; **mfdlg** – додає діалог функції приналежності; **mfdrag** – перетягування функції приналежності за допомогою миші; **popundo** – відновлює зі стеку останні зміни (відміняє останні дії); **pushundo** – передає поточні дані у стек відновлення; **savedlg** – діалог запису перед закриттям; **statmsg** – зображує повідомлення у полі статусу; **updtfis** – оновлює засоби графічного інтерфейсу Fuzzy Logic Toolbox; **wsdlg** – діалог "відкриття з" / "збереження до" робочої області.

ТЕМА 5. НЕЧІТКА КЛАСТЕРИЗАЦІЯ

Загальна характеристика задач кластерного аналізу

Терміном *кластерний аналіз* прийнято позначати сукупність методів, підходів і процедур, розроблених для вирішення проблеми формування однорідних класів в довільній проблемній області...

Нечіткий кластерний аналіз

Нечіткий кластерний аналіз використовується при побудові нейро-нечітких систем для визначення нечітких множин, якщо вони невідомі апріорі. Нечіткі множини знаходяться як проекції кластерів на кожну розмірність. Можливо поєднувати апріорні знання з кластерним аналізом, використовуючи його для уточнення параметрів функції приналежності...

Методи нечіткого кластерного аналізу. Метод FCM

Метод FCM (Fuzzy c-means – нечітких c-середніх) для вирішення задачі нечіткої кластеризації має ітеративний характер послідовного поліпшення певного вихідного нечіткого розбиття...

Метод гірської кластеризації

Метод пікового групування (гірської кластеризації), запропонований Р.Ягером та Д.Фільовим, не вимагає задавання кількості кластерів...

Метод поступово зростаючого розбиття

Метод поступово зростаючого розбиття (incremental decomposition algorithm) полягає в наступному. На першій ітерації є одне продукційне правило, що має як область свого впливу...

Загальна характеристика задач кластерного аналізу

Терміном *кластерний аналіз* прийнято позначати сукупність методів, підходів і процедур, розроблених для вирішення проблеми формування однорідних класів в довільній проблемній області.

Методи аналізу даних, складовою частиною яких є методи кластерного аналізу, не використовують апріорних припущень про імовірнісну природу вихідної інформації і керуються лише евристичними міркуваннями про характер і особливості досліджуваної сукупності об'єктів.

Кластерний аналіз (або автоматична класифікація, розпізнавання образів без вчителя) займає одне з центральних місць серед методів аналізу даних і є сукупністю підходів, методів і алгоритмів, призначених для знаходження деякого розбиття досліджуваної сукупності об'єктів на підмножини схожих між собою об'єктів. При цьому вихідним припущенням для виділення таких підмножин, що отримали спеціальну назву *кластерів,* служить лише неформальне припущення про те, що об'єкти, які відносяться до одного кластера, повинні мати більшу схожість між собою, чим з об'єктами з інших кластерів.

Вихідною інформацією для кластеризації є матриця спостережень:

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \cdots & & & \\ x_{S1} & x_{S2} & \cdots & x_{Sn} \end{bmatrix},$$

кожний рядок якої є значеннями n ознак одного з S об'єктів кластеризації. Кластеризація полягає в розбитті об'єктів з **X** на декілька підмножин (кластерів), в яких об'єкти між собою більш схожі, ніж з об'єктами з інших кластерів. У метричному просторі «схожість» зазвичай визначають через відстань. Відстань може розраховуватися як між вихідними об'єктами – рядками матриці X, так і від цих об'єктів до прототипів центрів кластерів. Часто координати прототипів заздалегідь невідомі, їх знаходять одночасно з розбиттям даних на кластери.

Кластерний аналіз (кластеризація) – це технологія, що дозволяє розподілити вхідні дані на *класи* – групи однотипних екземплярів вибірки, або *кластери* – компактні області групування екземплярів вибірки у просторі ознак. Вихідною інформацією для кластеризації є вибірка спостережень $x=\{x^s_j\}$, де x^s_j – значення j-ої ознаки s-го екземпляра вибірки, $s = 1, 2, \ldots, S$; $j=1, 2, \ldots, N$, S – кількість екземплярів вибірки, N – кількість ознак, що характеризують екземпляри вибірки.

Задача кластеризації полягає в розбитті об'єктів з x на декілька кластерів, у яких об'єкти більш схожі між собою, ніж з об'єктами інших кластерів. У метричному просторі «схожість» звичайно визначають через відстань.

Методи кластеризації можна класифікувати на чіткі та нечіткі. *Чіткі методи кластеризації* розбивають вихідну множину об'єктів x на декілька непересічних підмножин. При цьому будь-який об'єкт із x належить тільки одному кластеру.

Нечіткі методи кластерного аналізу дозволяють будь-якому екземпляру одночасно належати до всіх визначених кластерів, але з різним ступенем.

Концептуальний взаємозв'язок між кластерним аналізом і теорією нечітких множин заснована на тій обставині, що при вирішенні завдань структуризації складних систем більшість формованих класів об'єктів розмиті за своєю природою. Ця розмитість полягає в тому, що перехід від приналежності до неприналежності елементів до даних класів швидше поступовий, чим скачкоподібний. Тому найбільш адекватну відповідь в подібного роду випадках слід шукати не на питання: "Чи належить даний елемент тому або іншому класу?", а на питання: "У якій мірі даний елемент належить даному класу?".

Вимога знаходження однозначної кластеризації елементів досліджуваної проблемної області є досить грубою і жорсткою, особливо при рішенні задач системного аналізу, що слабо структуруються. Методи нечіткої кластеризації послабляють цю вимогу. Послаблення вимоги здійснюється за рахунок введення в розгляд нечітких кластерів і відповідних їм функцій приналежності, що набувають значень з інтервалу [0, 1].

В загальному випадку завданням нечіткої кластеризації є знаходження нечіткого розбиття множини елементів досліджуваної сукупності, які утворюють структуру нечітких кластерів, присутніх у вхідних даних. Це завдання зводиться до знаходження мір приналежності елементів універсуму шуканим нечітким кластерам, які в сукупності і визначають нечітке розбиття вихідної множини елементів.

Приклад 1. «Метелик» представляє собою 15 об’єктів, двовимірне зображення яких нагадує одноймённу комаху (рис. 1а). При чіткій кластеризації (рис. 1б і в) отримуємо два кластери з семи і восьми об’єктів.

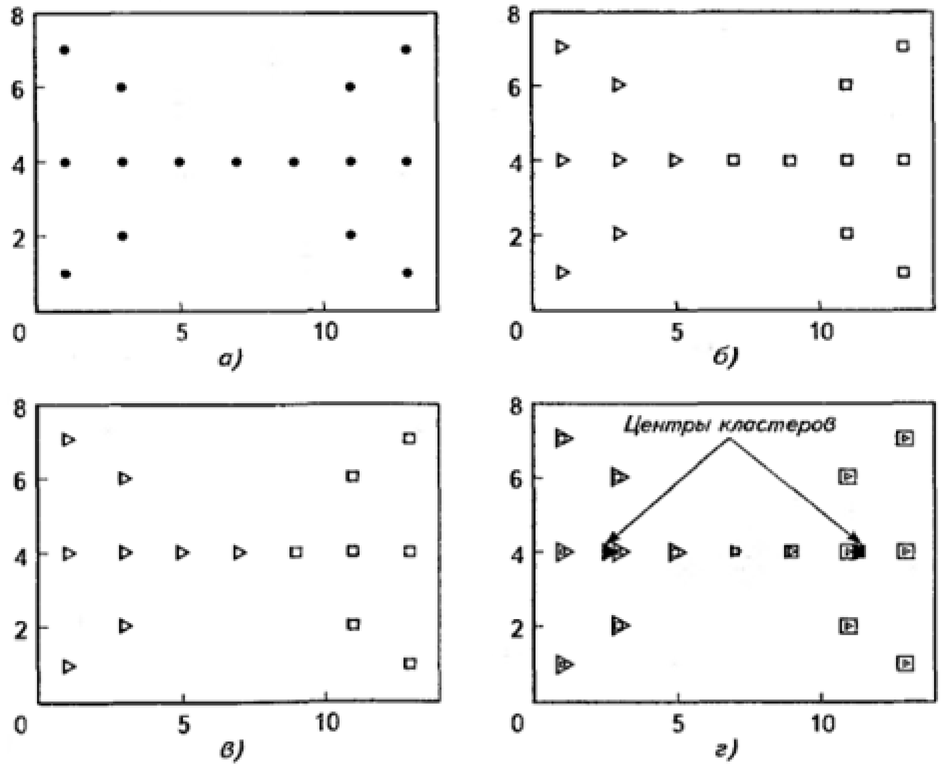


Рис. 1 – Порівняння чіткої та нечіткої кластеризації «метелика»

а – вхідні дані; б – чітка кластеризація I;

в – чітка кластеризація II; г – нечітка кластеризація

На рисунку об'єкти першого кластера позначені трикутниками, а другого – квадратами. Симетричний «метелик» при чіткій кластеризації розбивається на два несиметричні кластери. При нечіткій кластеризації (рис.6.1г) проблемний восьмий об'єкт, розташований в центрі «метелика», одночасно належить двом симетричним кластерам з однією і тією ж мірою. На цьому малюнку розмір маркерів пропорційний мірі приналежності об'єкта кластеру.

Нечіткий кластерний аналіз

Нечіткий кластерний аналіз використовується при побудові нейро-нечітких систем для визначення нечітких множин, якщо вони невідомі апріорі. Нечіткі множини знаходяться як проекції кластерів на кожну розмірність. Можливо поєднувати апріорні знання з кластерним аналізом, використовуючи його для уточнення параметрів функції приналежності. Недоліком такого методу визначення нечітких множин є складність їхньої інтерпретації.

Більшість методів нечіткої кластеризації спрямовані на мінімізацію суми:

$$J(x,u,C)=\sum_{s=1}^{S}\sum_{v=1}^{V}\left(\left(u_v^s\right)^m d^2\left(x^s,C^v\right)\right)$$

при виконанні умов:

$$V>1,\ \sum_{s=1}^{S}u_v^s>0,\ \sum_{v=1}^{V}u_v^s=1,$$

де S – кількість екземплярів, N – кількість параметрів, що описують один екземпляр (або кластер), V – кількість кластерів; $x=(x^1, x^2, ..., x^S)^T$ – це матриця входів для екземплярів навчаючої вибірки, $x^s = (x^s{}_1, x^s{}_2, ..., x^s{}_N)$ – входи s-го екземпляра, $s=1, 2,..., S$, $u = (u^1, u^2, ..., u^S)^T$ – матриця приналежностей екземплярів до кожного з кластерів, $u^s = (u^s{}_1, u^s{}_2, ..., u^s{}_V)$ – вектор приналежностей s-го екземпляра до кожного з кластерів, $u^s{}_v$Î[0,1], $C=(C^1,C^2, ...,C^V)^T$ – матриця центрів кластерів, $C^v = (C^v{}_1, C^v{}_2, ..., C^v{}_N)$ – центр v-

го кластера, $v = 1, 2, ..., V$, $m > 1$ – ступінь нечіткості отриманого розподілу (зазвичай обирається рівним 2), $d(x^s, C^v)$ – відстань між s-м екземпляром та центром v-го кластера.

Координати центрів кластерів визначають за формулою:

$$C_j^v = \frac{\sum_{s=1}^{S} \left(u_v^s\right)^m x_j^s}{\sum_{s=1}^{S} \left(u_v^s\right)^m}.$$

Найбільш простим є метод, в якому відстань між екземпляром та кластером знаходиться як *евклідова відстань*:

$$d(x^s, C^v) = \sqrt{\sum_{j=1}^{N} \left(x_j^s - C_j^v\right)^2}.$$

Такий метод шукає кластери як сфери однакового розміру.

Більш складні методи кластеризації шукають кластери як гіпереліпсоїди різного розміру. Такі методи називають *частковими*, вони не можуть вірно опрацьовувати шуми та викиди і віднаходити кластери з неопуклими поверхнями. Для проведення кластерного аналізу за допомогою часткового методу необхідно задати його параметри: діапазон значень змінних, кількість кластерів для кожної із змінних (або їх ширину), функцію приналежності, що описує кластери та інші параметри в залежності від обраного методу кластеризації.

За допомогою *ієрархічних методів* можна віднайти кластери, об'єднуючи менші кластери та розподіляючи більші. Таким чином знаходиться дерево кластерів, на різних рівнях якого можна отримати різне розподілення на кластери.

Щільнісні методи та *сіткові методи* дозволяють розподіляти на кластери різного розміру довільно розподілені екземпляри. Вони також добре впізнають шуми та викиди, але потребують ретельного вибору параметрів, необхідних для реалізації методу.

Множину ознак у об’єктів кластеризації слід вибирати так, щоб всі значення ознак були виміряні в шкалі відношень або шкалі інтервалів. У цьому випадку результати нечіткої кластеризації мають змістовну інтерпретацію, адекватну проблемі знаходження нечітких кластерів.

Інтервальна шкала. В процесі виміру ознаки об’єкту 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, вимірюваної в інтервальній шкалі: температура в шкалах Цельсія і Фаренгейта.

Шкала відношень. В процесі виміру ознаки об'єкту ставиться у відповідність також деяке дійсне число, рівне 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 в метрах і футах, маса в кілограмах і фунтах, швидкість в км/год і вузлах.

Метод FCM

Методи нечіткого кластерного аналізу. Метод FCM

Метод FCM (Fuzzy c-means – нечітких c-середніх) для вирішення задачі нечіткої кластеризації має ітеративний характер послідовного поліпшення певного вихідного нечіткого розбиття $R(A)=\{A_\nu | A_\nu \subseteq A\}$, що задається користувачем або формується автоматично за певним евристичним правилом. На кожній з ітерацій рекурентно переraховуються значення функцій приналежності нечітких кластерів та їхніх типових представників.

Метод FCM закінчить роботу у випадку, коли відбудеться виконання заданого апріорі деякого кінцевого числа ітерацій, або коли мінімальна абсолютна різниця між значеннями функцій приналежності на двох послідовних ітераціях не стане менше деякого апріорі заданого значення.

Крок 1. Попередньо необхідно задати такі значення: кількість шуканих нечітких кластерів V (V > 1), максимальну кількість ітерацій методу Epochs, параметр збіжності методу ε, а також експонентну вагу розрахунку цільової функції і центрів кластерів m (як правило, m = 2). Як поточне нечітке розбиття на першій ітерації методу для вибірки даних x задати деяке вихідне нечітке розбиття

$$R(A)=\{A_v | A_v \subseteq A\}$$

на V непорожніх нечітких кластерів, що описуються сукупністю функцій приналежності Usv, s = 1. 2...., S; v = 1, 2...., V. Нечітке розбиття отримують шляхом генерації випадковим чином елементів Uv. що задовольняють умовам цільової функції.

Крок 2. Для вихідного поточного нечіткого розбиття

$$R(A)=\{A_v | A_v \subseteq A\}$$

. розрахувати центри нечітких кластерів Cvj, v = 1, 2...., V; j=1, 2.....N та значення цільової функції J(x, u, C). Кількість виконаних ітерацій покласти рівною 1.

Крок 3. Сформувати нове нечітке розбиття

$$R'(A)=\{A_v | A_v \subseteq A\}$$

вихідної множини об'єктів кластеризації A на V непорожніх нечітких кластери, що характеризуються сукупністю функцій приналежності Us`v, v = 1. 2....,V, Xs є A, які визначаються за формулою:

$$u_v^{s}{}' = \left(\sum_{g=1}^{V} \left(\frac{d(x^s, C^v)}{d(x^s, C^g)} \right)^{\frac{2}{m-1}} \right)^{-1}, v = 1,2,...,V.$$

Крок 4. При цьому якщо для деякого v та деякого Xs значення d(Xs, Cv)=0, то для відповідного нечіткого кластера встановимо Us`v = 1, а для інших кластерів Uh`g = 0, g = 1,2....,V; h = 1,2.....,S. Якщо ж таких v для деякого Xs виявиться декілька, тобто для них значення d(Xs, Cv)=0, то евристично для меншого з v встановимо Us`v = 1. а для інших встановимо Uh`g = 0, g = 1,2….,V; h = 1,2….,S.

Крок 5. Для нового нечіткого розбиття

$$R'(A)=\{A_v|A_v \subseteq A\}$$

розрахувати центри нечітких кластерів Cvj та значення цільової функції J`(x,u,C).

Крок 6. Якщо кількість виконаних ітерацій перевищує задане число Epochs або ж модуль різниці |J(x,u,C)-J`(x,u,C)| $\leq \varepsilon$ . то за шуканий результат нечіткої клас-тернзації прийняти нечітке розбиття R'(A) і закінчити виконання методу. У протилежному випадку вважати поточним нечітким розбиттям R(A) = R`(A) і перейти до кроку 3 методу, збільшивши на 1 кількість виконаних ітерацій.

Вибір кількості кластерів V є однією з найважливіших проблем у розглянутому методі. Правильно вибрати кількість кластерів для реальних задач без будь-якої апріорної інформації про структуру даних досить складно. Існує два формальних підходи до вибору кількості кластерів.ε

Перший підхід заснований на критерії компактності та роздільності отриманих кластерів. Логічно припустити, що при правильному виборі кількості кластерів дані будуть розбиті на компактні і добре віддільні одна від іншої групи.

У іншому випадку , кластери, імовірно, не будуть компактними і добре віддільними.

Існує кілька критеріїв оцінки компактності кластерів , однак питання про те, як формально і вірогідно визначити правильність вибору кількості кластерів для довільного набору даних залишається відкритим. Для методу FCM рекомендується використовувати *індекс Хіє-Бені* (Xie-Beni index):

$$\chi = \frac{\sum_{v=1}^{V}\sum_{s=1}^{S}\left(u_v^s\right)^m d^2(x^s, C^v)}{S \min d^2(x^s, C^v)}.$$

Другий підхід заснований на *редукції кількості кластерів* і пропонує починати кластеризацію при досить великій кількості кластерів, а потім послідовно поєднувати схожі суміжні кластери.

Діагональна норма дозволяє виділяти кластери у вигляді гіпереліпсоїдів, орієнтованих уздовж координатних осей. При цьому використовуються різні формальні критерії схожості кластерів.Для діагональної норми матриця *B* задається у такий спосіб:

$$B_{i,j} = \begin{cases} w_i, i = j; \\ 0, i \neq j, \end{cases}$$

де w_i – елементи головної діагоналі матриці, що інтерпретуються як ваги координат.

Норма Махаланобіса дозволяє виділяти кластери у вигляді гіпереліпсоїдів, вісі яких можуть бути орієнтовані в довільних напрямках. Для норми Махалонобіса матриця *B* розраховується через коваріаційну матрицю від *x*:

$$B = G^{-1},\ G = \frac{1}{S}\sum_{s=1}^{S}\left(x^s - \overline{x}\right)^T\left(x^s - \overline{x}\right),\ \overline{x} = \frac{1}{S}\sum_{s=1}^{S} x^s,$$

де *G* – коваріаційна матриця; – вектор середніх значень даних.

Приклади ізоліній різних норм показані на рис.1.

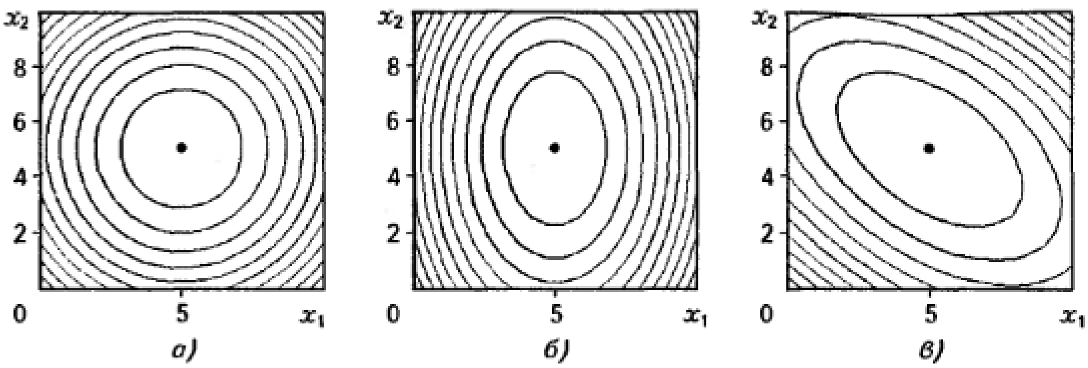


Рис.1 – Ізолінії різних норм

а – евклідова норма; *б* – діагональна норма; *в* – норма Махаланобіса

На рис. 2 наведений приклад кластеризації методом нечітких c-середніх при відстані Евкліда: зліва показані об'єкти кластеризації; справа показані результати нечіткої кластеризації. Центри нечітких кластерів позначені символами «+»

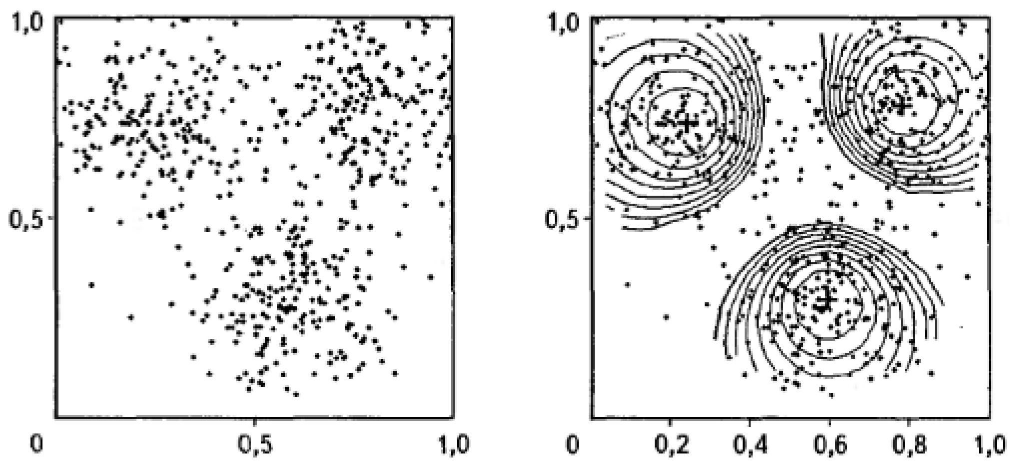


Рис.2 – Нечітка кластеризація при евклідовій нормі

Вісім ізоліній функцій приналежності нечітких кластерів побудовано для значень 0,67; 0,71, 0,75; 0,79; 0,83; 0,87; 0,91 і 0,95.

Для деяких наборів даних «очна кластеризація» дозволяє виділити скупчення об'єктів у вигляді різних геометричних фігур: сфер, еліпсоїдів різної орієнтації, ланцюжків і тому подібне.

У результаті застосування методів кластеризації з фіксованою нормою форма всіх кластерів виходить однаковою. Методи кластеризації ніби нав'язують даним невластиву їм структуру, що приводить не тільки до неоптимальних, але й іноді до принципово неправильних результатів. Для усунення цього недоліку запропоновано декілька методів, серед яких виділимо метод Густавсона-Кесселя.

Метод Густавсона-Кесселя має значно більшу обчислювальну трудомісткість у порівнянні з методом FCM.

Результати нечіткого кластер-аналізу можна використовувати для *синтезу нечітких правил*. Кожен кластер буде являти собою деяке нечітке правило, що узагальнює підмножину екземплярів навчаючої вибірки, найбільш тісно розташованих у просторі ознак. Функції приналежності термів у посилках правила отримують проектуванням ступенів приналежності відповідного кластера на вхідні змінні. Потім отримані множини ступенів приналежностей апроксимують придатними параметричними функціями приналежності. Як висновок правила сінглтонної бази знань вибирають координату центра кластера. Висновки правил бази знань Мамдані знаходять також як і функції приналежності термів вхідних змінних. Висновки правил бази знань Сугено знаходять за методом найменших квадратів. При кластеризації з використанням норми Махалонобіса як висновки правил типу Сугено можуть бути обрані рівняння довгих осей гіпереліпсоїдів.

Метод гірської кластеризації

У результаті застосування методів кластеризації з фіксованою нормою форма

Метод пікового групування (гірської кластеризації), запропонований Р.Ягером та Д.Фільовим, не вимагає задавання кількості кластерів. Кластеризація гірським методом не є нечіткою, однак, її часто використовують при синтезі нечітких правил з даних.

Ідея методу полягає в тому, що спочатку визначають точки, які можуть бути центрами кластерів. Далі для кожної такої точки розраховується значення потенціалу, що показує можливість формування кластера в її околиці. Чим щільніше розташовані об'єкти в околиці потенційного центра кластера, тим вище значення його потенціалу. Після цього ітераційно вибираються центри кластерів серед точок з максимальними потенціалами. Метод гірської кластеризації можна записати як послідовність таких кроків.

Крок 1. Сформувати потенційні центри кластерів, число яких Q повинно бути кінцевим. Центрами кластерів можуть бути об'єкти кластеризації -екземпляри вибірки x. тоді Q = S, де S - кількість екземплярів у вибірці x. Другий спосіб вибору потенційних центрів кластерів полягає в дискретизації простору вхідних ознак. Д.ля цього діапазони зміни вхіднігх ознак розбивають на кілька інтервалів. Проводячи через точки розбиття прямі, паралельні координатним осям, одержуємо «гратовий» гіперкуб. Вузли цих ґрат і будуть відповідати центрам потенційних кластерів. Позначимо через qr - кількість значень, що можуть приймати центри кластерів за r-ою координатою, г = 1,2...., N.

Тоді кількість можливих кластерів буде дорівнювати:

$$Q = \prod_{r=1}^{N} q_r$$

Крок 2. Розрахувати потенціал центрів кластерів за формулою:

$$P(C^q) = \sum_{s=1}^{S} e^{-\alpha d(C^q, x^s)}, \; q = 1, 2, \ldots, Q,$$

де $C^q = \{C^q_j\}$ – потенційний центр q-го кластера, C^q_j – значення j-ої ознаки для центра q-го кластера; α – позитивна константа, $d(C^q, x^s)$ – відстань (наприклад, евклідова) між потенційним центром кластера C^q та об'єктом кластеризації x^s.

У випадку, коли об'єкти кластеризації задані двома ознаками (N = 2), графічне зображення розподілу потенціалу буде являти собою поверхню, що нагадує гірський рельєф (рис. 1). Звідси і назва – гірський метод кластеризації.

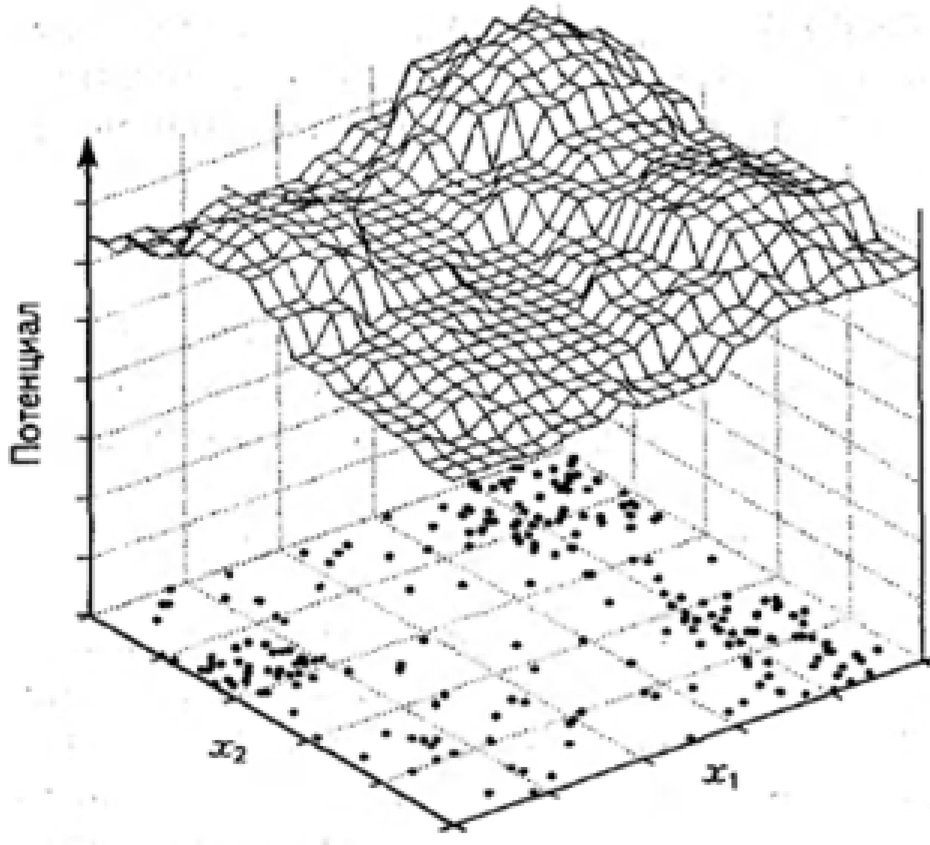


Рис. 1 - Розподіл потенціалу при кластеризації гірським алгоритмом

Крок 3. Вибрати як центри кластерів координати «гірських» вершин. Для цього, центром першого кластера призначають точку з найбільшим потенціалом. Звичайно, найвища вершина оточена декількома досить високими піками. Тому призначення центром наступного кластера точки з максимальним потенціалом серед вершин, що залишилися, призвело б до виділення великого числа близько розташованих центрів кластерів.

Щоб вибрати наступний центр кластера необхідно спочатку виключити вплив тільки що знайденого кластера. Для цього значення потенціалу для можливих центрів кластерів, що залишилися, перераховується в такий спосіб: від поточних значень потенціалу віднімають внесок центра тільки що знайденого кластера (тому кластеризацію за цим методом іноді називають субтрактивною).

Перерахунок потенціалу відбувається за формулою: $P_2(C^q)=P_1(C^q)-P_1(C^{v_1})e^{-\beta d(C^q,C^{v_1})}$, де P_1 – потенціал на 1-й ітерації; P_2 – потенціал на 2-й ітерації; v_1 – номер першого знайденого центра кластера: $v_1=\arg\max\limits_{q=1,2,...,Q} P_1(C^q)$, β – позитивна константа.

Номер центра другого кластера визначається за максимальним значенням оновленого потенціалу: $v_2=\arg\max\limits_{q=1,2,...,Q} P_2(C^q)$.

Потім знову перераховується значення потенціалів:

$$P_3(C^q)=P_2(C^q)-P_2(C^{v_2})e^{-\beta d(C^q,C^{v_2})}.$$

Крок 4. Якщо максимальне значення потенціалу перевищує деякий поріг, перейти до кроку 2, у противному випадку – зупинення. Метод гірської кластеризації є ефективним, якщо розмірність вхідного вектора не є занадто великою. У противному випадку (при великій кількості ознак) число потенційних центрів наростає лавинообразно, і процес розрахунку чергових пікових функцій стає занадто тривалим, а процедура кластеризації – малоефективною.

Метод гірської кластеризації можна використовувати для синтезу нечіткої бази знань. Нехай ми маємо $\{C^q\}$ – центри кластерів, знайдені в результаті гірської кластеризації, кожному з яких зіставлене значення цільової ознаки y^q. Тоді кожному центру кластера C^q ставиться у відповідність правило: Якщо $x^s\in C^q$, то $y^s\in y^q$, де нечіткі терми $x^s\in C^q$ та $y^s\in y^q$ характеризуються гаусівськими функціями приналежності:

$$\mu_{x^s\in C^q}=e^{-\frac{1}{2\beta}\sum\limits_{j=1}^{N}(C_j^q-x_j^s)^2},\quad \mu_{y^s\in y^q}=e^{-\frac{1}{2\beta}(y^q-y^s)^2}.$$

Синтез нечітких правил за результатами кластеризації

Нехай об’єкти кластеризації мають дві ознаки (*N*=2). Тоді результати нечіткого розбиття можна представити тривимірною поверхнею. Для побудови такої поверхні слід для кожного об’єкту відкласти по осях абсцис і ординат значення ознак, а по осі аплікат – міру приналежності об’єкта нечіткому кластеру. Кількість поверхонь дорівнює числу кластерів (*V*).

За результатами нечіткої кластеризації можна синтезувати нечіткі правила різних баз знань: синглтонної, Мамдані і Сугено. Функції приналежності термів в посилках правила отримують проектуванням ступеня приналежності відповідного кластера (рядків матриці нечіткого розбиття F) на осі вхідних змінних. Потім отримані множини ступенів приналежності апроксимують відповідними параметричними функціями приналежності. Як висновок правила синглтонної бази знань вибирають координату центру кластера. Висновки правил бази знань Мамдані знаходять так же як і функції приналежності термів вхідних змінних. Висновки правил бази знань Сугено знаходять по методу найменших квадратов. При кластеризації з використанням норми Махалонобіса як висновки правил типа Сугено можуть бути вибрані рівняння довгих осей гіпереліпсоїдів.

Вони інтерпретуються функціями приналежності термів, показаними на рис.6.9, за наступними нечіткими правилами:

ЯКЩО х = «низький», ТО у = «низький»

ЯКЩО x = «високий», ТО у = «високий». Функції приналежності отримані проектуванням поверхонь з рис.6.8 на осі x і y. Маркери на графіках функцій приналежності відповідають одному об’єкту кластеризації.

Метод субтрактивної кластеризації

Метод різницевого групування (субтрактивної кластеризації, subtractive clustering) на відміну від попереднього методу, як потенційні центри кластерів розглядає екземпляри навчаючої вибірки.

Метод поступово зростаючого розбиття

Метод поступово зростаючого розбиття (incremental decomposition algorithm) полягає в наступному. На першій ітерації є одне продукційне правило, що має як область свого впливу (область, у якій значення результуючої функції приналежності передумови нечіткого правила перевищує задану величину) усю множину припустимих вхідних значень (рис. 1).

На другій ітерації дане правило розбивається на два двома способами (показані стрілками). Проводиться навчання і вибирається, який зі способів розбиття дає найменшу погрішність (даний перехід відзначений чорною стрілкою). Серед наявних правил вибирається те, для якого складова погрішності в загальній погрішності є найбільшою (область його впливу заштриховано). Воно і підлягає розбиттю на два правила двома способами (ітерація 3). Описаний процес продовжується до досягнення необхідної точності або поки не буде згенеровано задане число продукційних правил.

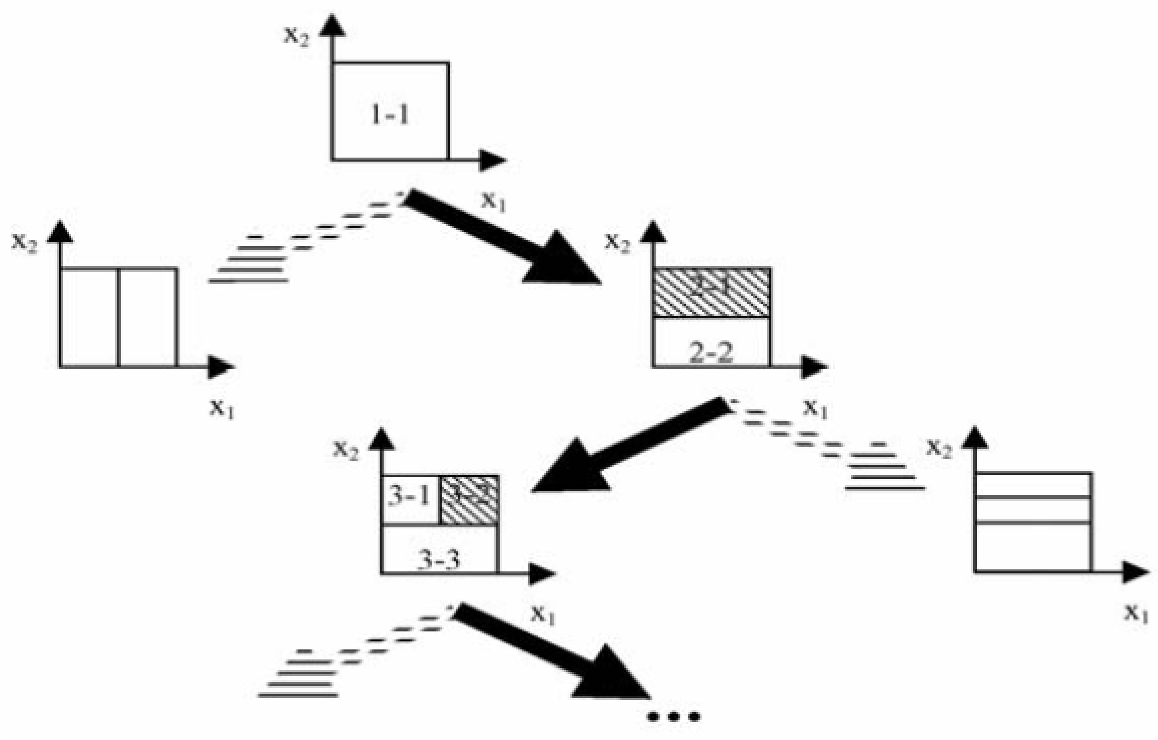


Рис.1 – Схема роботи методу IDA

ТЕМА 6. РОЗВ'ЯЗУВАННЯ ЗАДАЧ В MATLAB

Розв'язання задачі нечіткої кластеризації в командному вікні системи Matlab

У системі MATLAB для вирішення завдання нечіткої кластеризації на основі алгоритму FCM (Fuzzy c-means – нечітких с-середніх) може бути використана спеціальна функція командного рядка fcm або спеціальний графічний інтерфейс кластеризації, що викликається функцією findcluster...

Рішення задачі нечіткої кластеризації з використанням засобів графічного інтерфейсу

Функція findciuster призначена для виклику графічного інтерфейсу програми нечіткої кластеризації для методу нечітких с-середніх і методу субтрактивной кластеризації (subtractive clustering). Вона може бути викликана в одному з наступних форматів: findcluster або findcluster ('file.dat')...

Рішення задачі визначення числа кластерів для нечіткої кластеризації в системі Matlab

В разі відсутності яких-небудь апріорних припущень відносно кількості нечітких кластерів в системі MATLAB можна використовувати функцію командного рядка subclust або метод субтрактивної нечіткої кластеризації, реалізований в графічному інтерфейсі кластеризації...

Розв'язання задачі нечіткої кластеризації в командному вікні системи Matlab

системі MATLAB для вирішення завдання нечіткої кластеризації на основі алгоритму FCM (Fuzzy c-means – нечітких c-середніх) може бути використана спеціальна функція командного рядка fcm або спеціальний графічний інтерфейс кластеризації, що викликається функцією findcluster.

Функція командного рядка fem призначена для вирішення задачі нечіткої кластеризації з використанням алгоритму FCM. Вона може бути викликана в одному з наступних форматів:

[center, U, obj_fcn) = fcm(data, cluster_n) або

[center, U, obj_fcn] = fcm(data, cluster_n, options)

Вхіднимн аргументами цієї функції є:

- data: матриця початкових даних X кластеризації, s-рядок якої представляє інформацію про один об'єкт нечіткої кластеризації a_sÎA у формі вектора $x^s = (x^s{}_1, x^s{}_2, ..., x^s{}_N)$, $x^s{}_j$ – кількісне значення ознаки p_jÎP для об'єкта даних a_sÎA, s=1, 2,..., S –кількість екземплярів кластеризації, N – кількість параметрів (ознак), що описують один екземпляр (або кластер);

- cluster_n: кількість шуканих нечітких кластерів vÎV (більше одиниці).

Вихідними аргументами цієї функції є:

- center: матриця центрів шуканих нечітких кластерів $C =(C^1,C^2, ...,C^V)^T$, v = 1, 2, ..., V, кожен рядок якої представляє собою координати центру одного з нечітких кластерів у формі вектора $C^v = (C^v{}_1, C^v{}_2, ..., C^v{}_N)$;

- U: матриця значень функцій приналежності шуканого нечіткого розбиття $u^s = (u^s{}_1, u^s{}_2, ..., u^s{}_V)$, $u^s{}_v$Î[0,1];

obj_fcn: значення цільової функції $J(x,u,C)$ на кожній з ітерацій роботи алгоритму FCM.

Функція fcm(data, cluster_n, options) може бути викликана з додатковими аргументами options, які призначені для управління процесом нечіткої кластеризації, а також для зміни критерію зупинки роботи алгоритму і відображення інформації на екрані монітора.

Ці додаткові аргументи мають наступні значення:

- options (1) : експоненціальна вага m для розрахунку матриці нечіткого розбиття *U* (за умовчанням це значення дорівнює *m=2);*

- options (2): максимальне число ітерацій *k* (за умовчанням це значення дорівнює *k*=100);

- options (3): параметр збіжності алгоритму e (за умовчанням це значення дорівнює e=0.00001);

- options (4): інформація про поточну ітерацію, що відображується на екрані монітора (за умовчанням це значення дорівнює 1).

Якщо будь-яке із значень додаткових аргументів дорівнює NAN (не число), то для цього аргументу використовується значення за умовчанням. Функція fcm закінчує свою роботу, коли алгоритм FCM виконає максимальну кількість ітерацій *k,* або коли різниця між значеннями цільових функцій *J(x,u,C)* на двох послідовних ітераціях буде менше заданого апріорі значення параметра збіжності алгоритму e.

Функция fcm реализована в виде m-файла и использует, в свою очередь, три другие функции: функцию initfcm для формирования матрицы исходного разбиения некоторым случайным образом, функцию distfcm расчета матрицы расстояний между точками данных и центрами кластеров и функцию stepf cm для значений целевой функции и функций принадлежности объектов нечетким кластерам на каждой итерации работы алгоритма FCM. Все эти функции также реализованы в виде m-файлов и находятся в папке C:\MATLAB\toolbox\fuzzy\fuzzy.

Функція fcm реалізована у вигляді m-файла і використовує, у свою чергу, три інші функції: функцію initfcm для формування матриці вихідного розбиття деяким випадковим чином, функцію distfcm розрахунку матриці відстаней між точками даних і центрами кластерів і функцію stepfcm для розрахунку значень цільової функції і функцій приналежності об’єктів нечітким кластерам на кожній ітерації роботи алгоритму FCM. Всі ці функції також реалізовані у вигляді m-файлів і знаходяться в каталозі C:\Program

Files\Matlab\R2008a\toolbox\fuzzy\fuzzy.

Приклад 1

Розглянемо множину даних, що містяться в системі MATLAB і використовуються як тестова сукупність об'єктів нечіткої кластеризації. Ці дані представляють собою матрицю спостережень *X* розмірністю 140x2, яка міститься у файлі fcmdata.dat, що поставляється разом з системою MATLAB. В цьому випадку матриця спостережень *X* відповідає 140 об'єктам, для кожного з яких виконані виміри по двох ознаках, що є зручним для візуалізації вихідних даних і результатів нечіткої кластеризації в двовимірному просторі на площині. Графік координат точок на площині, відповідних об'єктам нечіткої кластеризації, представлений на рис. 1.

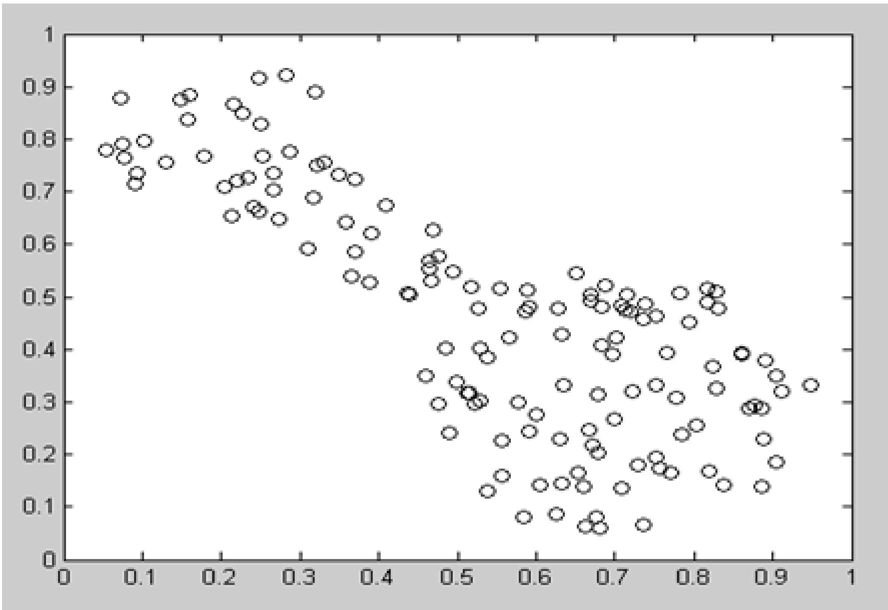


Рис. 1 – Множина об'єктів нечіткої кластеризації

У лістингу 1 наводиться послідовність команд, які забезпечують рішення задачі нечіткої кластеризації множини даних *X* і візуалізацію отриманих результатів. В даному прикладі використовується перший формат запису функції fcm.

Лістинг 1

```
load fcmdata.dat
plot(fcmdata(:, 1), fcmdata(:,2), 'o', 'color', 'k')
[center, U, obj_fcm] = fcm(fcmdata, 2);
maxU = max(U);
index1 = find(U(1, :) == maxU);
index2 = find(U(2, :)== maxU);
line(fcmdata(index1, 1), fcmdata(index1, 2), 'linestyle', 'none', 'marker', 'o', 'color', 'g');
line(fcmdata(index2, 1), fcmdata(index2, 2), 'linestyle', 'none', 'marker', 'x', 'color', 'r');
hold on
plot(center(1,1), center(1,2), 'ko', 'MarkerSize', 12, 'LineWidth', 2)
plot(center(2,1), center(2,2), 'kx', 'MarkerSize', 12, 'LineWidth', 2)
plot(obj_fcm)
```

Результат рішення задачі нечіткої кластеризації для двох нечітких кластерів з використанням вказаної послідовності команд може бути візуалізований (рис.2).

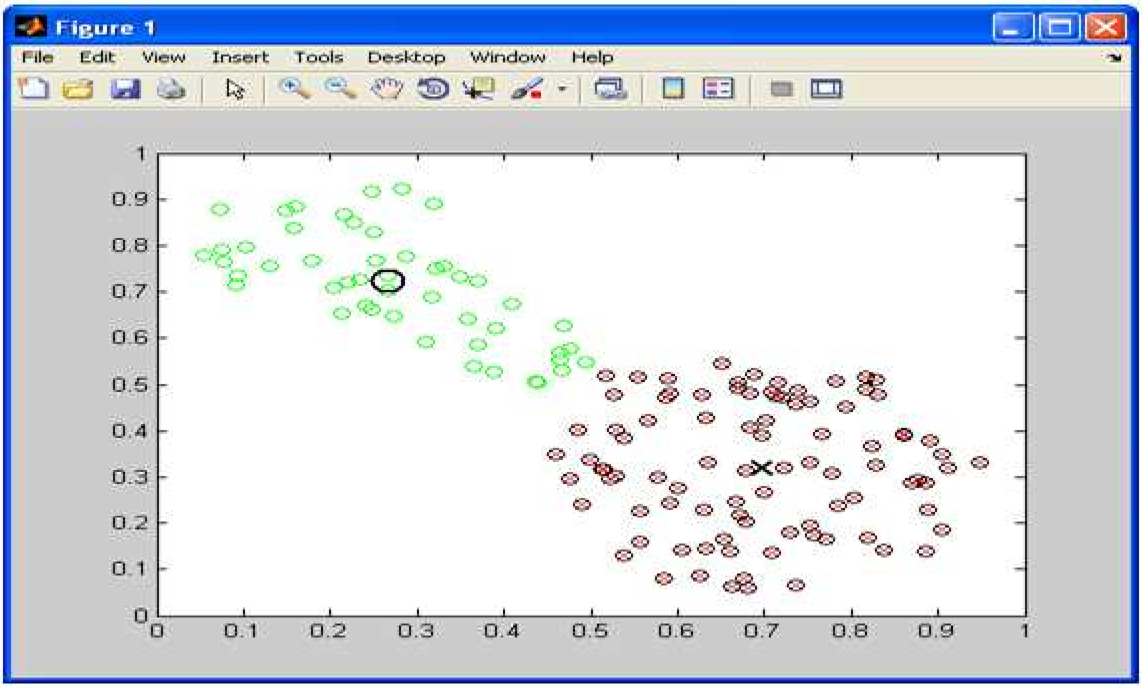


Рис.2 – Результат рішення задачі нечіткої кластеризації

Якщо після запису функції fcm в третьому рядку не вказувати крапку з комою (;), то у вікні команд відображуватимуться значення координат центрів нечітких кластерів, значення функцій приналежності об’єктів нечітким кластерам і значення цільової функції на кожній з ітерацій роботи алгоритму FCM (рис.3).

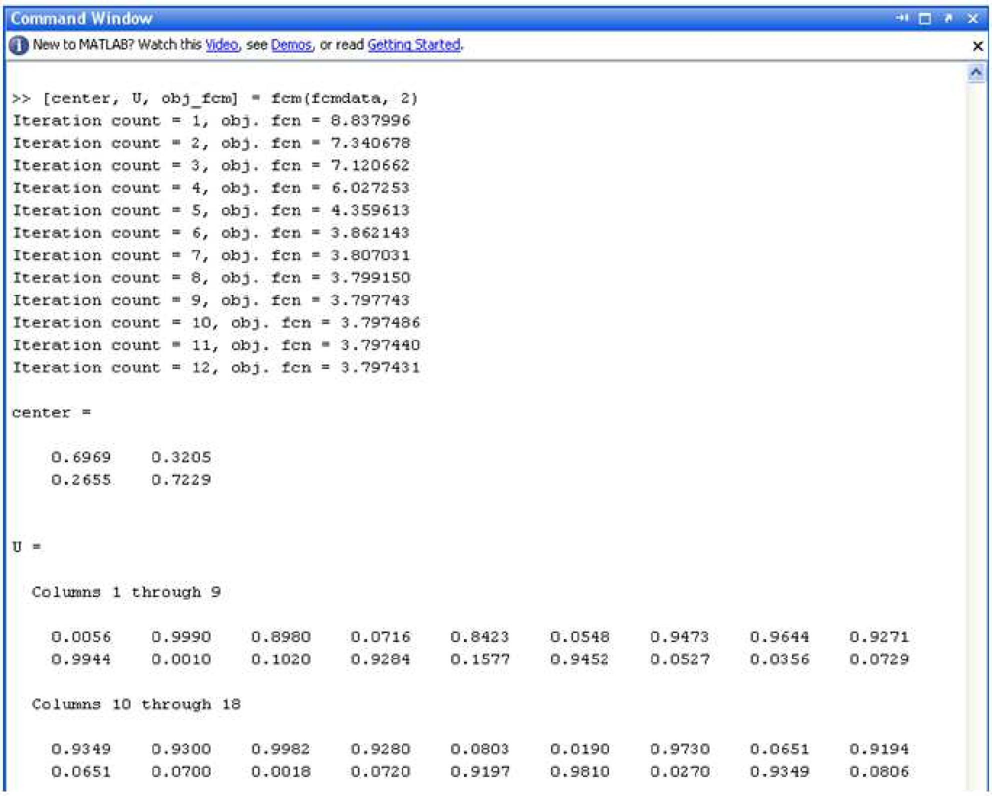


Рис.3 – Результат рішення задачі нечіткої кластеризації в командному вікні системи Matlab

Рішення задачі нечіткої кластеризації з використанням засобів графічного інтерфейсу

Функція **findciuster** призначена для виклику графічного інтерфейсу програми нечіткої кластеризації для методу нечітких с-середніх і методу субтрактивной кластеризації (subtractive clustering). Вона може бути викликана в одному з наступних форматів: findcluster або findcluster ('file.dat').

У першому випадку функція findcluster викликає графічний інтерфейс GUI програми для виконання нечіткої кластеризації алгоритмом FCM і нечіткої субтрактивної кластеризації (рис.1). При цьому необхідно завантажити в робочу область вихідні дані із зовнішнього файлу за допомогою кнопки **Load Data** (рис.2).

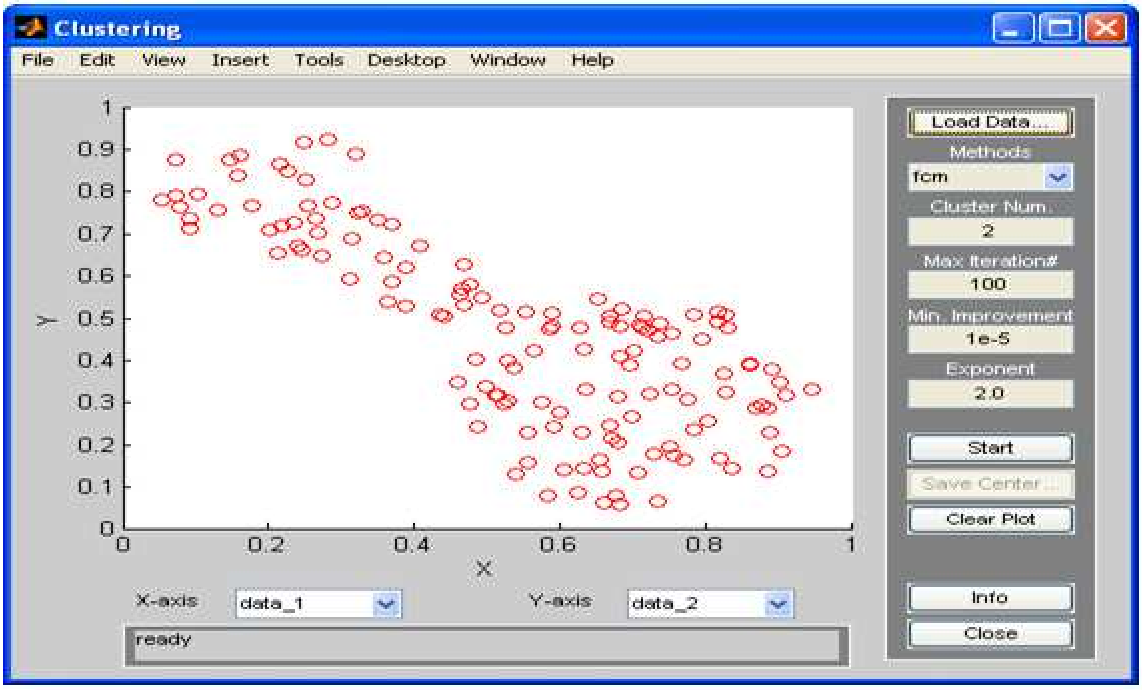


Рис.1 – Графічний інтерфейс GUI програми для виконання нечіткої кластеризації алгоритмом FCM і нечіткої субтрактивної кластеризації

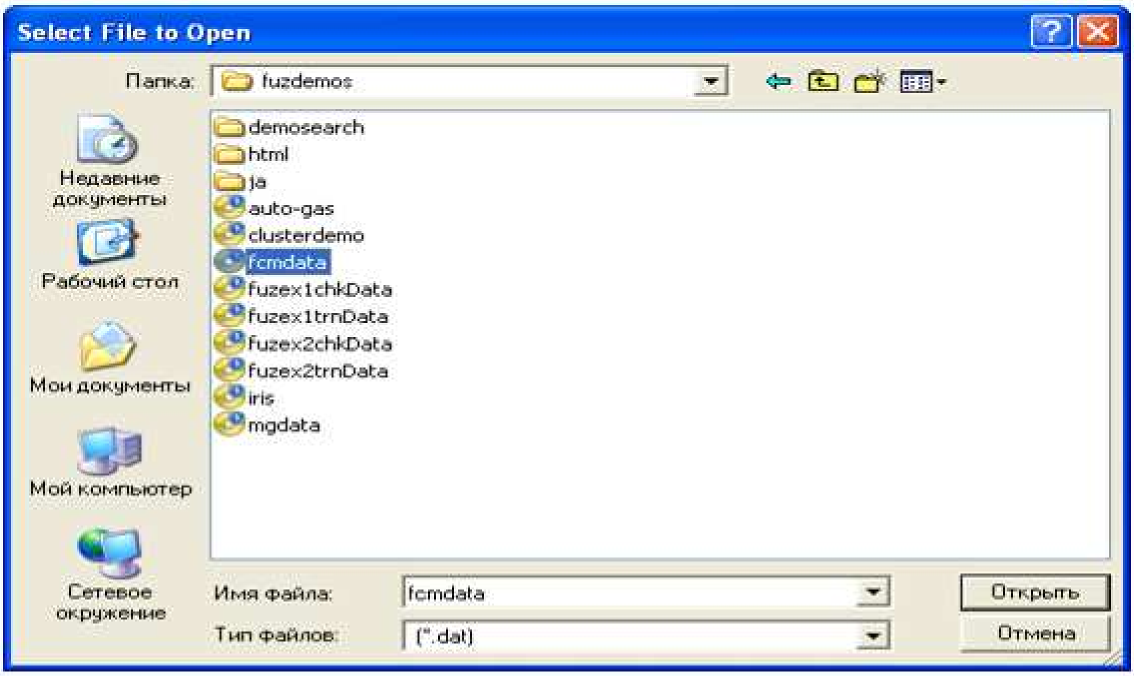


Рис.2 – Вибір зовнішнього файлу для завантаження в робочу область

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 умовчанням. Ці значення можуть бути змінені користувачем. Для цього необхідно встановити курсор введення у відповідне поле і набрати потрібні цифри з врахуванням допустимих значень параметрів. Призначення цих параметрів для алгоритму FCM описане раніше при розгляді функції fcm. Призначення цих параметрів для алгоритму субтрактивной кластеризації буде описано нижче при розгляді функції subclust.

У другому випадку функція findclusterl ('file.dat') викликає графічний інтерфейс, а в робочу обрасть автоматично завантажуються дані кластеризації із зовнішнього файлу file.dat. При цьому на графіку відображуються значення матриці даних для перших двох ознак (рис.1).

Після натиснення на кнопку **Start** починає роботу відповідний алгоритм нечіткої кластеризації з параметрами, встановленими за умовчанням або зміненими користувачем. Результати роботи алгоритму відображуються на графіку (рис.3).

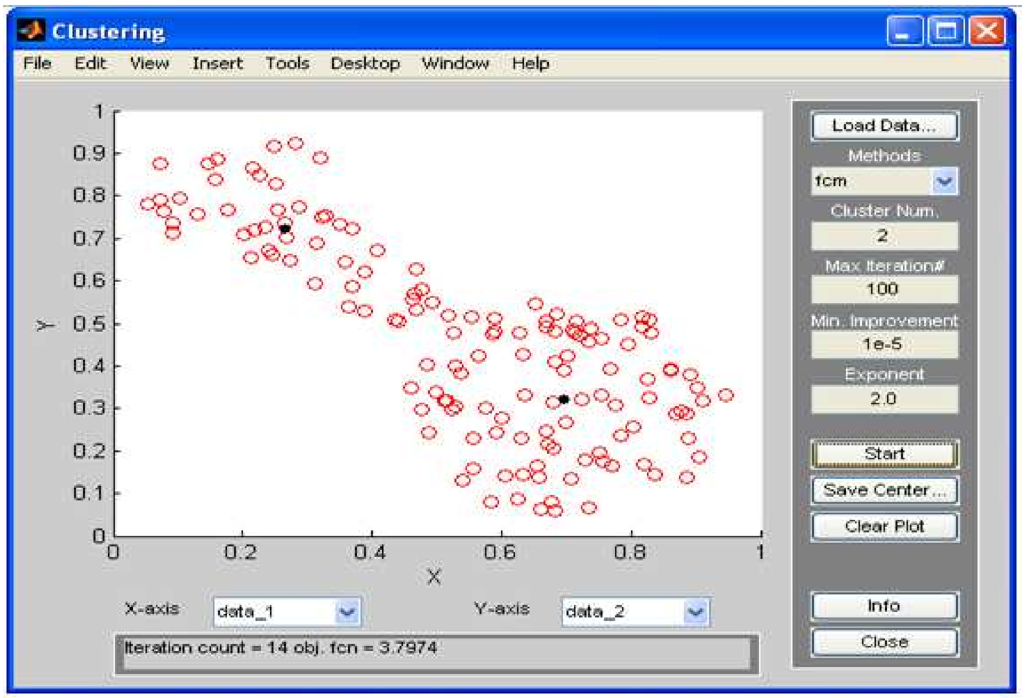


Рис.3 – Результат рішення задачі нечіткої кластеризації алгоритмом FCM

Знайдені центри кластерів змальовані чорними кругами і їх координати можна зберегти в зовнішньому файлі з метою подальшого аналізу.

Інколи число нечітких кластерів, необхідних для роботи алгоритму FCM, апріорі є невідомим. В цьому випадку доцільно використовувати реалізований в системі MATLAB так званий алгоритм субтрактивної кластеризації.

Рішення задачі визначення числа кластерів в системі Matlab

В разі відсутності яких-небудь апріорних припущень відносно кількості нечітких кластерів в 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 як центр потенційного кластера, після чого слід обчислити деяку міру здатності кожної точки даних представляти центр кластера. Ця кількісна міра заснована на оцінці щільності точок даних довкола відповідного центру кластера.

Даний алгоритм, який є узагальненням методу кластеризації Р.Ягера (R.Yager), заснований на виконанні наступних дій:

1. Вибрати точку даних з максимальним потенціалом для представлення центру першого кластера.

2. Видалити всі точки даних в околиці центру першого кластера, величина якої задається параметром radii, аби визначити наступний нечіткий кластер і координати його центру.

Ці дві процедури повторюються до тих пір, поки всі точки даних не виявляться всередині околиць радіусу radii шуканих центрів кластерів.

Функція командного рядка subclust знаходить центри кластерів методом субтрактивної кластеризації. Вона використовується в наступному форматі:

[C, S] = subclust (X, radii, xBounds, options)

При цьому матриця *X* містить дані кластеризації. Кожен рядок матриці

X відповідає координатам окремої точки даних. Параметр radii є вектором, компоненты якою набувають значень з інтервалу [0, 1] і задають діапазон розрахунку центрів кластерів по кожній з ознак вимірів. При цьому робиться припущення, що всі дані містяться в деякому одиничному гіперкубі. У загальному випадку малі значення параметра radii призводять до знаходження малого числа великих по кількості точок кластерів. Найкращі результати виходять при значеннях radii між 0.2 і 0.5.

Аргумент xBounds є матрицею розмірності (2´*q*), яка визначає спосіб відображення матриці даних *X* в деякому одиничному гіперкубі. Тут *q* – кількість ознак, що розглядаються в множині даних. Цей аргумент є необов’язковим, якщо матриця *X* вже нормалізована. Перший рядок матриці xBounds містить мінімальні значення інтервалу виміру кожної з ознак, а другий рядок – максимальні значення виміру кожної з ознак.

Для зміни заданих за умовчанням параметрів алгоритму кластеризації може бути використаний додатковий вектор options. Компоненти цього вектора можуть набувати наступних значень:

- options (1) = quashFactor – параметр, використовуваний як коефіцієнт для множення значень radii, які визначають околицю центру кластера. Це здійснюється з метою зменшення впливу потенціалу граничних точок, що розглядаються як частина нечіткого кластера (за умовчанням це значення дорівнює 1.25);

- options (2) = acceptRatio – параметр, що встановлює потенціал як частину потенціалу центру першого кластера, вище за який інша точка даних може розглядатися як центр іншого кластера (за умовчанням це значення дорівнює 0.5);

- options (3) = rejectRatio — параметр, що встановлює потенціал як частину потенціалу центру першого кластера, нижче за який інша точка даних не може розглядатися як центр іншого кластера (за умовчанням це значення дорівнює 0.15);

- options(4) = verbose – якщо значення цього параметра не дорівнює нулю, то на екран монітора виводиться інформація про виконання процесу кластеризації (за умовчанням це значення дорівнює 0).

Функція subclust повертає матрицю *C* значень координат центрів нечітких кластерів. При цьому кожен рядок цієї матриці містить координати одного центру кластера. Ця функція також повертає вектор *S*, компоненти якого представляють значення s, які визначають діапазон впливу центру кластера по кожній з ознак, що розглядаються. При цьому всі центри кластерів володіють однаковою множиною значень s.

Приклад 1.

Розглянемо рішення задачі визначення кількості кластерів для множини вихідних даних з прикладу 7.1. Нагадаємо, що ці дані є матрицею даних *X* розмірністю (140x2), яка міститься у файлі fcmdata.dat. У цьому випадку виклик функції субтрактивної кластеризації може бути реалізований таким чином:

```
load fcmdata.dat
[C,S] = subclust(fcmdata, [0.5 0.5], [], [1.25 0.5 0.15 1])
```

Оскільки для множини даних з прикладу 7.1 використовується 2 ознаки (матриця datin має 5 стовпців), то по кожній з ознак задаються радіуси околиць 0.5 і 0.5 відповідно. Коефіцієнти шкал для відображення множини даних в одиничний гіперкуб находять на основі мінімального і максимального значень даних.

Аргумент squashFactor = 1.25 вказує на те, що необхідно визначити кластери, недалеко розташовані один від одного. Аргумент acceptRatio=0.5 вказує на те, що для знаходження центрів кластерів дуже високий потенціал не є необхідним. Аргумент rejectRatio=0.15 не виключає з розгляду точки даних, що не володіють високим потенціалом.

Нарешті, останній аргумент verbose = 1 дозволяє виведення інформації про виконання процесу кластеризації на екран монітора.

В результаті виконання цього фрагмента команд будуть отримані наступні значення матриці центрів кластерів і вектора s-значень (рис.1).

```
>> load fcmdata.dat
[C,S] = subclust(fcmdata, [0.5 0.5], [], [1.25 0.5 0.15 1])
Normalizing data...
Computing potential for each data point...
Found cluster 1, potential = 1
Found cluster 2, potential = 0.603359
Found cluster 3, potential = 0.271582

C =

    0.6962    0.3893
    0.2661    0.7364
    0.6614    0.0641


S =

    0.1580    0.1525
```

Рис.1 – Результат рішення задачі нечіткої субтрактивної кластеризації в командному вікні системи Matlab

Як можна відмітити, для вказаних значень аргументів функція субтрактивної кластеризації, що розглядається, знаходить три нечіткі кластери і відображує координати їх центрів в командному вікні системи MATLAB.

Для вирішення завдання субтрактивної кластеризації може бути також використаний розглянутий вище графічний інтерфейс нечіткої кластеризації, який викликається функцією findcluster. Виклик цієї функції може бути здійснений в одному з наступних форматів: findcluster або findcluster ('file.dat'), особливості яких були викладені раніше і ілюстровані на рис.7.4-7.6.

Приклад 2.

Розглянемо рішення задачі визначення кількості кластерів для безлічі вихідних даних з прикладу 7.1 з використанням графічного інтерфейсу кластеризації. Для цього завантажимо вихідні дані із зовнішнього файлу командою findcluster ('fcmdata.dat'), виберемо метод кластеризації subtractive в списку Methods і натиснемо кнопку Start. Інші параметри залишимо запропонованими за замовчуванням. Результат розв'язання задачі субтрактівної кластеризації зображений на рис.7.8 і також містить 3 нечітких кластера.

Розглянемо рішення задачі визначення кількості кластерів для множини вихідних даних з прикладу 7.1 з використанням графічного інтерфейсу кластеризації. Для цього завантажимо вихідні дані із зовнішнього файлу командою findcluster ('fcmdata.dat'), виберемо метод кластеризації subtractive в списку **Methods**, що розкривається, і натискуватимемо кнопку **Start.** Останні параметри залишимо запропонованими за умовчанням. Результат рішення задачі субтрактивної кластеризації представлений на рис.2 і також містить 3 нечітких кластера.

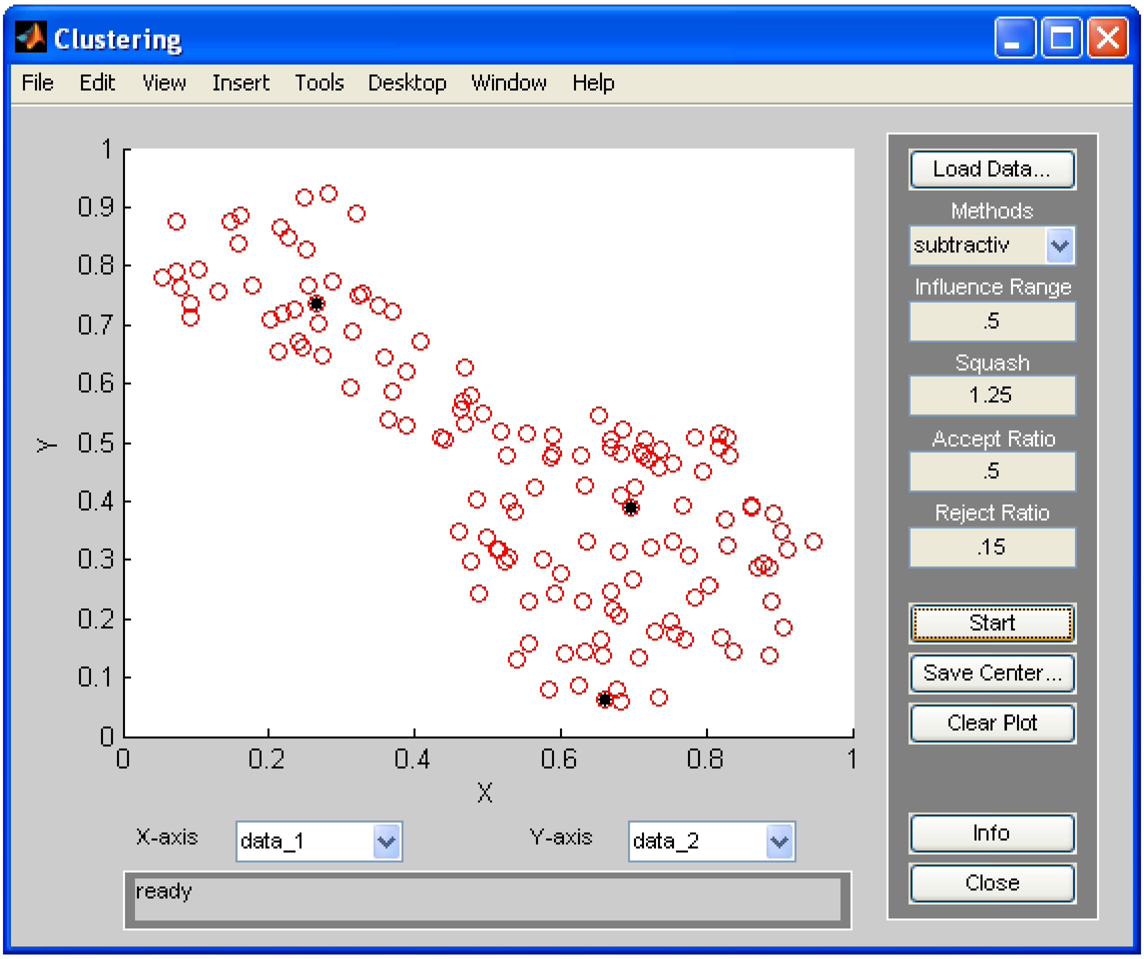


Рис.2 – Результат рішення задачі нечіткої субтрактивної кластеризації з використанням графічного інтерфейсу кластеризації

Таким чином, система MATLAB дозволяє вирішувати завдання нечіткої кластеризації двома способами: за допомогою функцій командного рядка і за допомогою графічного інтерфейсу кластеризації. Перший з них є більш трудомістким, але володіє більшою гнучкістю і можливістю відображення у вікні команд значень матриць центрів нечітких кластерів, функцій приналежності і цільової функції. Другий спосіб представляється найбільш зручним для виконання деякої серії розрахунків для різних значень вхідних параметрів з метою візуального аналізу отриманих результатів.

Результати нечіткої кластеризації мають наближений характер і можуть служити лише для попередньої структуризації інформації, що міститься в множині вихідних даних. Вирішуючи завдання нечіткої кластеризації, потрібно пам’ятати про особливості і обмеження процесу виміру ознак у сукупності об’єктів кластеризації. Оскільки нечіткі кластери формуються на основі метрики Евкліда, відповідний простір ознак повинен задовольняти аксіомам метричного простору. В той же час для пошуку закономірностей в проблемній області, що мають неметричний характер, необхідно використовувати спеціальні засоби і інструментарій, розроблені для інтелектуального аналізу даних (Data Mining).

ТЕМА 7. НЕЙРО-НЕЧІТКІ МЕРЕЖІ

Загальна характеристика та властивості нейро-нечітких мереж

Різні типи інтелектуальних систем мають свої особливості, наприклад, за можливостями навчання, узагальнення і отримання результатів, що робить їх найбільш придатними для вирішення одних класів задач і менш придатними – для інших...

Формування бази знань нейро-нечіткої мережі

Копіювання навчаючої вибірки в базу знань – для кожного екземпляра навчаючої вибірки формується окреме правило...

Елементи нейро-нечітких мереж

Систему нечіткого логічного висновку можна представити у вигляді *нейро-нечіткої мережі* – нейронної мережі прямого поширення сигналу особливого типу. Архітектура нейро-нечіткої мережі изоморфна нечіткій базі знань. В інтегрованій моделі для визначення параметрів системи нечіткого висновку використовуються методи навчання нейронних мереж...

Інтегровані нейро-нечіткі мережі

Систему нечіткого логічного висновку можна представити у вигляді *нейро-нечіткої мережі* – нейронної мережі прямого поширення сигналу особливого типу. Архітектура нейро-нечіткої мережі изоморфна нечіткій базі знань. В інтегрованій моделі для визначення параметрів системи нечіткого висновку використовуються методи навчання нейронних мереж. Інтегровані нейро-нечіткі системи розподіляють структури даних і подання знань...

Загальна характеристика та властивості нейро-нечітких мереж

Різні типи інтелектуальних систем мають свої особливості, наприклад, за можливостями навчання, узагальнення і отримання результатів, що робить їх найбільш придатними для вирішення одних класів задач і менш придатними – для інших.

Нейронні мережі, наприклад, є зручними для задач розпізнавання образів, але дуже незручні для пояснення, як вони таке розпізнавання здійснюють. Вони можуть автоматично здобувати знання, але процес їхнього навчання найчастіше відбувається досить повільно, а аналіз навченої мережі є дуже складним (навчена мережа є звичайно «чорною скринею» для користувача). При цьому яку-небудь апріорну інформацію (знання експерта) для прискорення процесу навчання в нейронну мережу ввести складно.

Системи з нечіткою логікою, напроти, є зручними для пояснення одержуваних за їхньою допомогою висновків, але вони не можуть автоматично здобувати знання для використання їх у механізмах виведень. Необхідність розбиття універсальних множин на окремі області, як правило, обмежує кількість вхідних змінних у таких системах невеликим значенням.

Хаяши (Y.Hayashi) та Імура (A.Imura) показали, що нейромережа прямого поширення може апроксимувати будь-яку систему, що заснована на нечітких правилах, та будь-яка нейромережа прямого поширення може бути апроксимована системою, що заснована на нечітких правилах.

Теоретично системи з нечіткою логікою і штучні нейронні мережі подібні одні одним, однак на практиці в них є власні переваги і недоліки. Дане розуміння лягло в основу створення апарату *нечітких нейронних мереж*, у яких виведення робляться на основі апарату нечіткої логіки, але відповідні функції приналежності настроюються з використанням методів навчання нейронних мереж, наприклад, методу зворотного поширення помилки. Такі системи не тільки використовують апріорну інформацію, але можуть здобувати нові знання, будучи *логічно прозорими*.

Нейро-нечітка мережа – це подання системи нечіткого виведення у вигляді нейронної мережі, зручної для навчання, аналізу та використання. Структура нейро-нечіткої мережі відповідає основним блокам систем нечіткого виведення.

Основними *властивостями нейро-нечітких мереж* є те, що:

– нейро-нечіткі мережі засновані на нечітких системах, які навчаються за допомогою методів, використовуваних у нейромережах;

– нейро-нечітка мережа зазвичай є багатошаровою (частіше – тришаровою) нейронною мережею. *Перший шар* становить вхідні змінні, *середній* становить нечіткі правила, а *третій* – вихідні змінні. Ваги підключення відповідають нечітким множинам вхідних і вихідних змінних. Іноді використовується п’ятишарова архітектура. В загальному випадку нечітка система необов’язково повинна бути подана в такому вигляді, однак це є зручною моделлю для застосування навчаючих методів;

– нейро-нечітка мережа завжди (до, під час і після навчання) може бути інтерпретована як система нечітких правил;

– процедура навчання враховує семантичні властивості нечіткої системи. Це виражається в обмеженні можливих модифікацій, які застосовуються до параметрів, що налагоджуються. Потрібно, однак, сказати, що не всі методи навчання нейро-нечітких мереж враховують семантику системи;

– нейро-нечітка система апроксимує *N´M* – розмірну невідому функцію, що частково описана навчаючими даними.

Типи поєднання нечіткої логіки і нейронних мереж за способом взаємодії виділяють такі:

– *нечіткі нейронні системи* (fuzzy neural systems). В цьому випадку в нейронних мережах застосовуються принципи нечіткої логіки для прискорення процесу налагодження або поліпшення інших параметрів. При такому підході нечітка логіка є лише інструментом нейронних мереж і така система не може бути інтерпретована в нечіткі правила, оскільки являє собою «чорну скриню»;

– *конкуруючі нейро-нечіткі системи* (concurrent neuro-fuzzy systems).

У таких моделях нечітка система і нейронна мережа працюють над однією задачею, не впливаючи на параметри одна одної. Можлива послідовна обробка даних спочатку однією системою, потім іншою;

– *паралельні нейро-нечіткі системи* (cooperative neuro-fuzzy systems).

У таких системах налагодження параметрів виконується за допомогою нейронних мереж. Далі нечітка система функціонує самостійно. Виділяють такі *типи паралельних нейро-нечітких моделей*: 1) нечітка асоціативна пам'ять (fuzzy associative memory); 2) системи із виділенням нечітких правил шляхом використання карт, що самоорганізуються (fuzzy rule extraction using selforganizing maps); системи, здатні навчати параметри нечітких множин (systems capable of learning fuzzy set parameters);

– *інтегровані* (*гібридні*) *нейро-нечіткі системи* (integrated neuro-fuzzy systems) – системи з тісною взаємодією нечіткої логіки і нейронних мереж.

Під терміном «нейро-нечіткі мережі» частіше за все мають на увазі системи саме цього типу. Як правило інтегровані системи є системами типу Мамдані або Такагі-Сугено.

За способом відображення нечітких множин в структурі мережі нейро-нечіткі мережі бувають трьох основних типів:

– *системи, побудовані на вибіркових нечітких множинах*. В таких системах ступені приналежності описані лише для деяких значень з області визначення і функція приналежності подана у вигляді вектору. Кожному ступеню приналежності відповідає лише один вхідний або вихідний нейрон. Існує два підходи до реалізації таких систем. В першому система просто апроксимує відповідність виходів входам, така система є «чорною скринею». В другому створюється система із спеціальною архітектурою, в якій втілюються нечіткі правила;

– *системи, параметризовані функції приналежності* яких зберігаються в нейронах. Прикладом таких систем є ANFIS (Adaptive-Network-based Fuzzy Inference System);

– *системи, в яких параметризовані функції приналежності використовуються як ваги зв'язків між нейронами.* Таку систему інакше можна назвати персептроном з нечіткими зв'язками або *нечітким персептроном*.

За характером навчання виділяють такі *типи нейро-нечітких мереж*:

– *самоналагоджувані нейро-нечіткі мережі* – з адаптацією структури та параметрів;

– *адаптивні нейро-нечіткі мережі* – із жорсткою структурою та адаптацією параметрів мережі.

Адаптивні нейро-нечіткі мережі *за видом методу оптимізації* поділяють на такі, що використовують детерміновані методи типу градієнтного пошуку, та такі, що використовують стохастичні методи, зокрема еволюційні.

Адаптивні нейро-нечіткі мережі *за типом параметрів адаптації* поділяють на мережі з адаптацією параметрів функцій приналежності, мереж із адаптацією ваг правил та мережі з адаптацією параметрів оператора агрегації.

Формування бази знань нейро-нечіткої мережі

Розглянемо об'єкт виду $y = f(x_1, x_2, \ldots, x_n)$ для якого зв'язок «входи x_j – вихід y» можна подати у вигляді *експертної матриці знань* (табл.8.1).

Таблиця 8.1 – Експертна матриця знань

Номер правила	Якщо (входи)				То (вихід)	Вага правила
	x_1	x_2	…	x_n		
11	a_1^{11}	a_2^{11}	…	a_n^{11}	d_1	w_{11}
12	a_1^{12}	a_2^{12}	…	a_n^{12}		w_{12}
…	…	…	…	…		…
$1k_1$	$a_1^{1k_1}$	$a_2^{1k_1}$	…	$a_n^{1k_1}$		w_{1k_1}
…	…	…	…	…	…	…
$m1$	a_1^{m1}	a_2^{m1}	…	a_n^{m1}	d_m	w_{m1}
$m2$	a_1^{m2}	a_2^{m2}	…	a_n^{m2}		w_{m2}
…	…	…	…	…		…
mk_m	$a_1^{mk_m}$	$a_2^{mk_m}$	…	$a_n^{mk_m}$		w_{mk_m}

Цій матриці відповідає нечітка база знань:

$$\text{Якщо } [(x_1 = a_1^{j1}) \text{ та } (x_2 = a_2^{j1}) \text{ та} \ldots (x_n = a_n^{j1})] \text{ (з вагою } w_{j1}) \ldots$$

$$\ldots \text{ або } [(x_1 = a_1^{jk_j}) \text{ та } (x_2 = a_2^{jk_j}) \text{ та} \ldots (x_n = a_n^{jk_j})] \text{ (з вагою } w_{jk_j}),$$

$$\text{то } \quad y = d_j, \quad j = 1, 2, ..., m, \text{p} = 1, 2, \ldots, k_j,$$

де a_i^{jp} – лінгвістичний терм, що оцінює змінну x_i у рядку p; p = 1, 2,…,k_j; k_j – кількість рядків-кон'юнкцій, що відповідають класу d_j вихідної змінної y, w_{jp} – число в діапазоні [0, 1], що характеризує суб'єктивну міру впевненості експерта щодо висловлення з номером p.

Якщо вихідна змінна (консеквент) є дискретною, то класи d_j, j = 1, 2, ..., m, формуються як номери дискретних значень вихідної змінної.

Якщо вихідна змінна є дійсною, то класи d_j, j = 1, 2, ..., m, формуються шляхом квантування діапазону $[\underline{y}, \overline{y}]$ вихідної змінної y на m рівнів:

$$[\underline{y}, \overline{y}] = \underbrace{[\underline{y}, y_1]}_{d_1} \cup \ldots \cup \underbrace{[y_{j-1}, y_j]}_{d_j} \cup \ldots \cup \underbrace{[y_{m-1}, \overline{y}]}_{d_m}.$$

Основні принципи формування бази знань нейро-нечітких систем полягають у наступному.

Копіювання навчаючої вибірки в базу знань – для кожного екземпляра навчаючої вибірки формується окреме правило. Перевагою даного методу є простота та висока швидкість роботи, недоліком – відсутність узагальнюючих властивостей і громіздкість одержуваної мережі.

Оптимізація кількості продукційних правил – знаходження такого значення кількості продукційних правил S, при якому значення помилки $E(S)$ є мінімальним, для чого при різних значеннях S навчають мережу і вимірюють значення помилки, після чого оптимізують функцію $E(S)$ за параметром *S*. Недоліком даного методу є дуже високі вимоги до обчислювальних ресурсів, обумовлені необхідністю заново навчати мережу на кожному кроці.

Спільна оптимізація ваг мережі та кількості продукційних правил шляхом вирішення багатоекстремальної оптимізаційної задачі або автоматичне визначення числа кластерів у навчаючій вибірці та встановлення центрів функцій приналежності в їхні центри на основі кластер-аналізу.

Скорочення (*редукція*) *правил.* В методах скорочення при ініціалізації формується нечітка система, що містить свідомо надлишкове число продукційних правил. У процесі роботи методу зайві продукційні правила виключаються.

Основні принципи редукції правил:

– скорочення нечітких правил відповідно до їхніх логічних функцій:

1) виключення правил, для яких результуюча функція приналежності менше визначеного порога, як таких, що мало впливають на остаточний результат;

2) виключення суперечливих правил, які взаємно компенсуються;

3) виключення одного з двох співпадаючих правил, як таких, що не несуть нової інформації;

– ортогоналізація: видалення тих продукційних правил, вплив яких на точність виявляється мінімальним після оцінки індивідуального внеску кожного продукційного правила у вихідний сигнал мережі, одержуваної шляхом використання ортогонального методу найменших квадратів.

Істотним недоліком методів скорочення є необхідність спочатку працювати зі свідомо надлишковою за розміром базою знань, що обумовлює в ряді випадків повільну роботу методів.

Елементи нейро-нечітких мереж

Розглянемо просту нейронну мережу, що складається з одного нейрона з двома входами (рис.8.1).

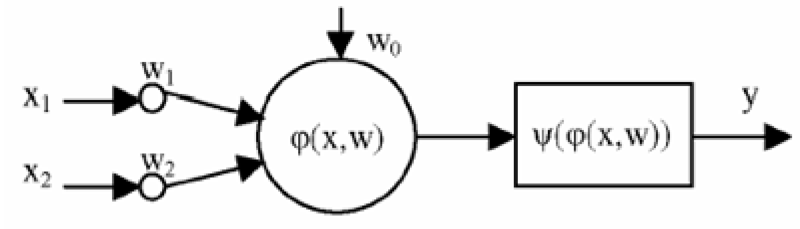


Рис.8.1 – Схема штучного нейрона

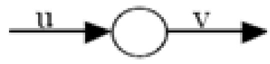
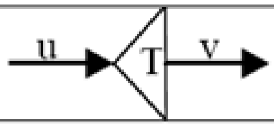
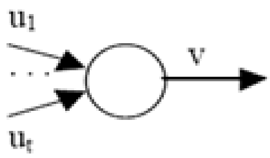
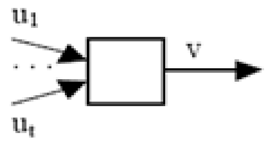
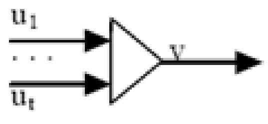
Вхідні сигнали x_i «взаємодіють» із *синаптичними вагами* w_i, утворюючи значення *рівня збудження нейрона* $\varphi(x,w) = w_1x_1+w_2x_2+w_0$, де $\varphi(x,w)$ – *функція постсинаптичного потенціалу* (дискримінантна, вагова функція) нейрона, w_0 – *поріг нейрона*. Вихід нейрона у утворюється в результаті перетворення значення $\varphi(x,w)$ *функцією активації* ψ:

$$y = \psi(\varphi(x,w)) = \psi(w_1 x_1 + w_2 x_2+w_0).$$

Розглянута однонейронна мережа, у якій використовуються операції множення, підсумовування і функція активації є *стандартною нейронною мережею*. У випадку застосування замість операцій множення, підсумовування й функцій активації таких операцій, як *t*-норма і *t*-конорма дану нейронну мережу будемо називати *нечіткою*. Входи, виходи і ваги нечіткої нейронної мережі – дійсні числа, що належать відрізку [0, 1].

У табл.8.2 наведено опис основних елементів нечітких нейромереж.

Таблиця 8.2 – Елементи нечітких нейромереж.

Назва елемента	Зображення елемента	Функція елемента
Вхід		$v = u$
Нечіткий терм		$v = \mu_T(u)$
Нечітке правило		$v = \min\limits_{i=1,2,\ldots,t} u_i$ $v = \prod\limits_{i=1}^{t} u_i$
Клас правил		$v = \max\limits_{i=1,2,\ldots,t} u_i$ або $v = \sum\limits_{i=1}^{t} u_i$
Дефазифікація		$v = \sum\limits_{j=1}^{m} u_j d_j \Big/ \sum\limits_{j=1}^{m} u_j$

Тут використовуються такі позначення: $\mu_T(x)$ – функція приналежності змінної x терму T; d_j – центр j-го класу правил, $d_j \in [\underline{y}, \overline{y}]$, де $\underline{y}$ та $\overline{y}$ – нижня і верхня межі значень вихідної змінної.

Розглянемо приклади *елементів нечітких нейронних мереж.*

Нечіткий нейрон «ТА». Сигнали x_j та ваги w_j у даному випадку поєднуються за допомогою t-конорми: $p_j = S(w_j, x_j)$, $j = 1, 2$, а вихід утворюється з застосуванням t-норми: $y = \text{AND}(p_1, p_2) = T(p_1, p_2) = T(S(w_1, x_1), S(w_2, x_2))$. Якщо прийняти $T = \min$, $S = \max$, то нечіткий нейрон «ТА» реалізує композицію min-max: $y = \min(\max(w_1, x_1), \max(w_2, x_2))$.

Нечіткий нейрон «АБО». Сигнали x_j і ваги w_j тут поєднуються за допомогою t-норми: $p_j = T(w_j, x_j)$, $j = 1, 2$, а вихід утворюється з застосуванням t-конорми: $y = \text{OR}(p_1, p_2) = S(p_1, p_2) = S(T(w_1, x_1), T(w_2, x_2))$. Якщо прийняти $T = \min$, $S = \max$, то нечіткий нейрон «АБО» реалізує композицію max-min: $y = \max(\min(w_1, x_1), \min(w_2, x_2))$.

Інтегровані нейро-нечіткі мережі

Систему нечіткого логічного висновку можна представити у вигляді *нейро-нечіткої мережі* – нейронної мережі прямого поширення сигналу особливого типу. Архітектура нейро-нечіткої мережі изоморфна нечіткій базі знань. В інтегрованій моделі для визначення параметрів системи нечіткого висновку використовуються методи навчання нейронних мереж. Інтегровані нейро-нечіткі системи розподіляють структури даних і подання знань.

Найбільш загальний спосіб застосування методу навчання до нечіткої системи полягає в тому, щоб подати її у вигляді архітектури, подібної нейронній мережі. Однак звичайні (градієнтні) методи навчання нейронних мереж не можуть безпосередньо застосовуватися до такої системи, оскільки функції, використовувані в процесі виведення звичайно є недиференційовані. Ця проблема може бути вирішена шляхом використання диференційованих функцій у системі виведення або не використання стандартних методів навчання нейромереж. У нейро-нечітких мережах використовуються реалізації трикутних норм (множення і імовірнісне АБО), що диференціюються, а також гладкі функції приналежності. Це дозволяє застосовувати для налаштування нейро-нечітких мереж швидкі алгоритми навчання нейронних мереж, засновані на методі зворотного поширення помилки.

Інтегрована нейро-нечітка модель є інтерпретованою і здатна до контрольованого навчання. У ANFIS процес навчання сконцентрований тільки на настроюванні значень параметрів у межах установлених структур. Для багатомірних задач буде складним визначити оптимальні структури «передумова-наслідок», кількість правил і т. д. Користувач повинний визначити деталі архітектури: тип і кількість функцій приналежності для вхідної і вихідної змінних, тип нечітких операторів і т. д.

Серед інтегрованих нейро-нечітких моделей ANFIS має найбільшу точність. Це пояснюється тим, що в ANFIS реалізовані правила Такагі-Сугено. Системи виведення типу Такагі-Сугено є більш точними, але вимагають більше обчислювальних витрат.

Правила Такагі-Сугено є іншим типом нечітких правил та мають такий вигляд: Якщо $x_1 \in A_1$ та $x_2 \in A_2$ та … та $x_n \in A_n$, то $y = f(x_1, x_2, \ldots, x_n)$. В правилах такої форми $\{x_i\}$ – це вхідні змінні; y – вихідна змінна; $\{A_i\}$ – нечіткі терми, визначені на $\{x_i\}$, $f(x_1, x_2, \ldots, x_n)$ – лінійна функція, що залежить від вхідних змінних.

Антецедентом правила називають множину нечітких термів, визначених для входів, що є умовою спрацьовування правила.

Консеквентом правила називають множину нечітких термів, визначених для виходів, які будуть присвоєні вихідним змінним при спрацьовуванні правила.

Імплікація – це модифікація нечітких множин виходів за допомогою ступеню виконання правила.

ANFIS реалізує систему нечіткого виведення Сугено у вигляді п'ятишарової нейронної мережі прямого поширення сигналу. Призначення шарів наступне:

• *перший шар* – терми вхідних змінних;

• *другий шар* – антецеденти (посилки) нечітких правил;

• *третій шар* – нормалізація мір виконання правил;

• *четвертий шар* – висновки правил;

• *п'ятий шар* – агрегація результату, отриманого за різними правилами.

Входи мережі в окремий шар не виділяються.

На рис.8.2 представлена ANFIS-мережа з двома вхідними змінними (x_1 і x_2) і чотирма нечіткими правилами. Для лінгвістичної оцінки вхідної змінної x_1 використовується три терми, для змінної x_2 – два терми.

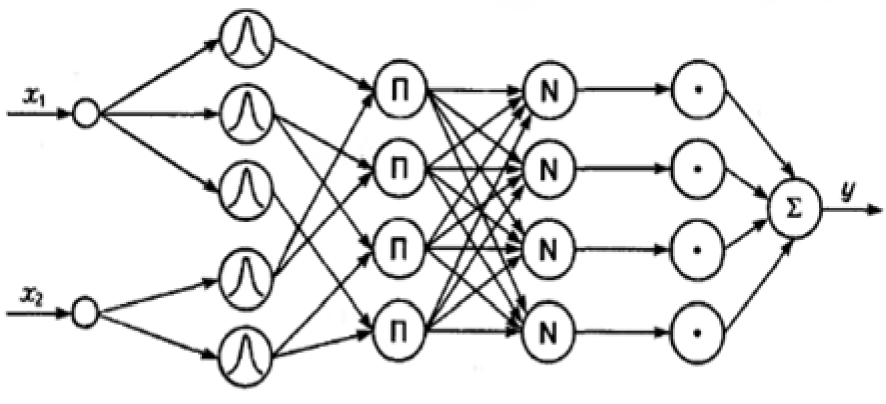


Рис.8.1 – Приклад нейро-нечіткої мережі

Введемо наступні позначення:

• $x_1, x_2, \ldots x_n$ – входи мережі;

• y – вихід мережі.

• R_r: Якщо $x_1=a_{1,r}$ ТА…ТА $x_n=a_{n,r}$, то $y=b_{0,r} + b_{0,r}x_1 + \ldots + b_{n,r}x_n$ – нечітке правило з порядковим номером r;

• m – кількість правил, ;

• $a_{i,r}$ – нечіткий терм з функцією приналежності $m_r(x_i)$, що застосовується для лінгвістичної оцінки змінної x_i в r-му правилі (,);

• $b_{q,r}$ – коефіцієнти в висновку r-го правила (,).

ANFIS-мережа функціонує таким чином.

Шар 1. Кожен вузол першого шару представляє один терм з колоколоподібною функцією приналежності. Входи мережі x_1, x_2,…x_n сполучені лише зі своїми термами. Кількість вузлів першого шару дорівнює сумі потужностей терм-множин вхідних змінних. На вихід вузла подається міра приналежності значення вхідної змінної відповідному нечіткому терму:

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

де a, b і c – параметри функції приналежності, що настроюються.

Шар 2. Кількість вузлів другого шару дорівнює m. Кожен вузол цього шару відповідає одному нечіткому правилу. Вузол другого шару сполучений з тими вузлами першого шару, які формують антецеденти відповідного правила. Отже, кожен вузол другого шару може приймати від 1 до n сигналів. Виходом вузла є міра виконання правила, яка розраховується як добуток вхідних сигналів. Позначимо виходи вузлів цього шару через t_r, .

Шар 3. Кількість вузлів третього шару також дорівнює m. Кожен вузол цього шару розраховує відносну міру виконання нечіткого правила по формулі:

$$\tau_r^* = \frac{\tau_r}{\sum\limits_{j=\overline{1,m}} \tau_j}.$$

Шар 4. Кількість вузлів четвертого шару також дорівнює m. Кожен вузол сполучений з одним вузлом третього шару, а також зі всіма входами мережі (на рис.8.2 зв'язки з входами не показані). Вузол четвертого шару розраховує вклад одного нечіткого правила у вихід мережі по такій формулі:

$$y_r = \tau_r^*(b_{0,r} + b_{1,r}x_1 + \ldots + b_{n,r}x_n).$$

Шар 5. Єдиний вузол цього шару підсумовує вклади всіх правил:

$$y = y_1 + \ldots y_r \ldots + y_m \,.$$

Типові процедури навчання нейронних мереж можуть бути застосовані для налаштування ANFIS-мережі, оскільки в ній використовуються функції, що диференціюються. Зазвичай застосовується комбінація градієнтного спуску у вигляді алгоритму зворотного поширення помилки і методу найменших квадратів. Алгоритм зворотного поширення помилки настроює параметри антецедентів правил, тобто функцій приналежності. Методом найменших квадратів оцінюються коефіцієнти висновків правил, оскільки вони лінійно пов'язані з виходом мережі. Кожна ітерація процедури настройки виконується в два етапи. На першому етапі на входи подається навчальна вибірка і по нев'язці між бажаною і дійсною поведінкою мережі методом найменших квадратів знаходяться оптимальні параметри вузлів четвертого шару. На другому етапі залишкова нев'язка передається з виходу мережі на входи і методом зворотного поширення помилки модифікуються параметри вузлів першого шару. При цьому знайдені на попередньому етапі коефіцієнти висновків правил не змінюються. Ітераційна процедура налаштування продовжується, поки нев'язка перевищує заздалегідь встановлене значення. Для настройки функцій приналежності, окрім методу зворотного поширення помилки, можуть використовуватися і інші алгоритми оптимізації, наприклад, метод Льовенберга-Марквардта.

ТЕМА 8. БАЗИ ЗНАНЬ МАМДАНІ І СУДЖЕНО

ІДЕНТИФІКАЦІЯ НЕЛІНІЙНИХ ЗАЛЕЖНОСТЕЙ НЕЧІТКИМИ БАЗАМИ ЗНАНЬ

Настройка нечіткої бази знань Мамдані

Ідентифікація нелінійних залежностей, тобто побудова їх моделей за результатами спостережень, є важливим завданням в техніці, економіці, політиці, медицині і в інших областях. Роботи по нечіткій ідентифікації нелінійних залежностей інтенсивно проводяться за кордоном з 1990-х років. Серед російськомовних публікацій виділимо роботи професора Ротштейна, в яких розроблений метод двохетапної ідентифікації нелінійних залежностей за допомогою нечітких баз знань. На першому етапі виконується структурна ідентифікація...

https://uk.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D0%BD%D0%B5%D1%87%D1%96%D1%82%D0%BA%D0%BE%D0%B3%D0%BE_%D0%B2%D0%B8%D0%B2%D0%B5%D0%B4%D0%B5%D0%BD%D0%BD%D1%8F#%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_%D0%9C%D0%B0%D0%BC%D0%B4%D0%B0%D0%BD%D1%96_(Mamdani)

Настройка нечіткої бази знань Суджено

Згідно з методом найменших квадратів, настройка нечіткої бази знань Суджено зводиться до наступної задачі математичного програмування: *знайти такий вектор (P, B), щоб...*

https://uk.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D0%BD%D0%B5%D1%87%D1%96%D1%82%D0%BA%D0%BE%D0%B3%D0%BE_%D0%B2%D0%B8%D0%B2%D0%B5%D0%B4%D0%B5%D0%BD%D0%BD%D1%8F#%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_%D0%A1%D1%83%D0%B3%D0%B5%D0%BD%D0%BE_(Sugeno)

Настройка нечіткої бази знань

Настройка нечіткого класифікатора є знаходженням таких параметрів функцій приналежностей термів вхідних змінних і вагових коефіцієнтів правил, які мінімізують відхилення між бажаною і дійсною поведінкою нечіткою класифікатора на навчальній вибірці. Критерій близькості можна визначити різними способами...

https://uk.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D0%BD%D0%B5%D1%87%D1%96%D1%82%D0%BA%D0%BE%D0%B3%D0%BE_%D0%B2%D0%B8%D0%B2%D0%B5%D0%B4%D0%B5%D0%BD%D0%BD%D1%8F#%D0%9E%D1%81%D0%BD%D0%BE%D0%B2%D0%BD%D1%96_%D0%B0%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC%D0%B8_%D0%BD%D0%B5%D1%87%D1%96%D1%82%D0%BA%D0%BE%D0%B3%D0%BE_%D0%B2%D0%B8%D0%B2%D0%B5%D0%B4%D0%B5%D0%BD%D0%BD%D1%8F

Настройка нечіткої бази знань Мамдані

Ідентифікація нелінійних залежностей, тобто побудова їх моделей за результатами спостережень, є важливим завданням в техніці, економіці, політиці, медицині і в інших областях. Роботи по нечіткій ідентифікації нелінійних залежностей інтенсивно проводяться за кордоном з 1990-х років. Серед російськомовних публікацій виділимо роботи професора Ротштейна, в яких розроблений метод двохетапної ідентифікації нелінійних залежностей за допомогою нечітких баз знань. На першому етапі виконується структурна ідентифікація. Вона є формуванням нечіткої бази знань, яка грубо відображає нелінійний взаємозв’язок «входи - вихід» за допомогою лінгвістичних правил <Якщо-то>. Ці правила генеруються експертом або отримуються в результаті екстракції нечітких знань з експериментальних даних. На другому етапі відбувається параметрична ідентифікація досліджуваної залежності шляхом знаходження таких параметрів нечіткої бази знань, які мінімізують відхилення результатів нечіткого моделювання від експериментальних даних. Параметрами, що настроюються, є ваги правил і параметри функцій приналежності нечітких термів.

Передбачається, що модель залежності y = f(X) задана нечіткою базою знань Мамдані. Вважатимемо, що існує також навчальна вибірка з M пар експериментальних даних, що зв'язують входи $X_r = (x_{r,1}, x_{r,2}, ..., x_{r,n})$ з виходом y досліджуваної залежності:

$$(X_r, y_r), \quad r = \overline{1, M},$$

де $X_r = (x_{r,1}, x_{r,2}, ..., x_{r,n})$ – вхідний вектор в r-й парі навчальної вибірки і y_r – відповідний вихід.

Введемо наступні позначення:

• P – вектор параметрів функцій приналежності термів вхідних і вихідної змінних;

• W – вектор вагових коефіцієнтів правил бази знань;

• $F(P, W, X_r)$ – результат виведення по нечіткій базі знань Мамдані з параметрами (P, W) при значенні входів X_r.

Згідно з методом найменших квадратів, настройка нечіткої бази знань Мамдані зводиться до наступної задачі математичного програмування: знайти такий вектор (P, W), щоб

$$\text{RMSE} = \sqrt{\frac{1}{M} \sum_{r=\overline{1,M}} (y_r - F(P, W, X_r))^2} \rightarrow \min.$$

У цьому завданні оптимізації на керовані змінні P зазвичай накладають обмеження, що забезпечують лінійну впорядкованість елементів терм-множин. Такі обмеження не дозволяють алгоритмам оптимізації зробити, наприклад, нечітку множину «низький» більше «високого». Крім того, ядра нечітких множин не повинні виходити за межі діапазонів зміни відповідних змінних. Такими обмеженнями забезпечується прозорість нечіткої бази знань після настройки, тобто можливість змістовної інтерпретації правил. Що стосується вектора W, то його координати повинні знаходитися в діапазоні [0, 1]. Якщо до рівня інтерпретабельності бази знань пред’являються високі вимоги, то ваги правил не настроюють, залишаючи їх рівними 1. Можливий і проміжний варіант, коли вагові коефіцієнти можуть набувати значень 0 або 1.

В цьому випадку нульове значення вагового коефіцієнта еквівалентне виключенню правила з нечіткої бази знань.

Завдання (9.2) може бути вирішено різними методами оптимізації, серед яких часто застосовується метод найскорішого спуску, квазіньютонівські методи і генетичні алгоритми.

Приклад 9.1. Результати спостережень залежності $y = f(x_1, x_2)$ представлені трьохмірним графіком (рис.9.1) і навчальною вибіркою з 20 пар «входи-вихід» (табл.9.1, рис.9.2).

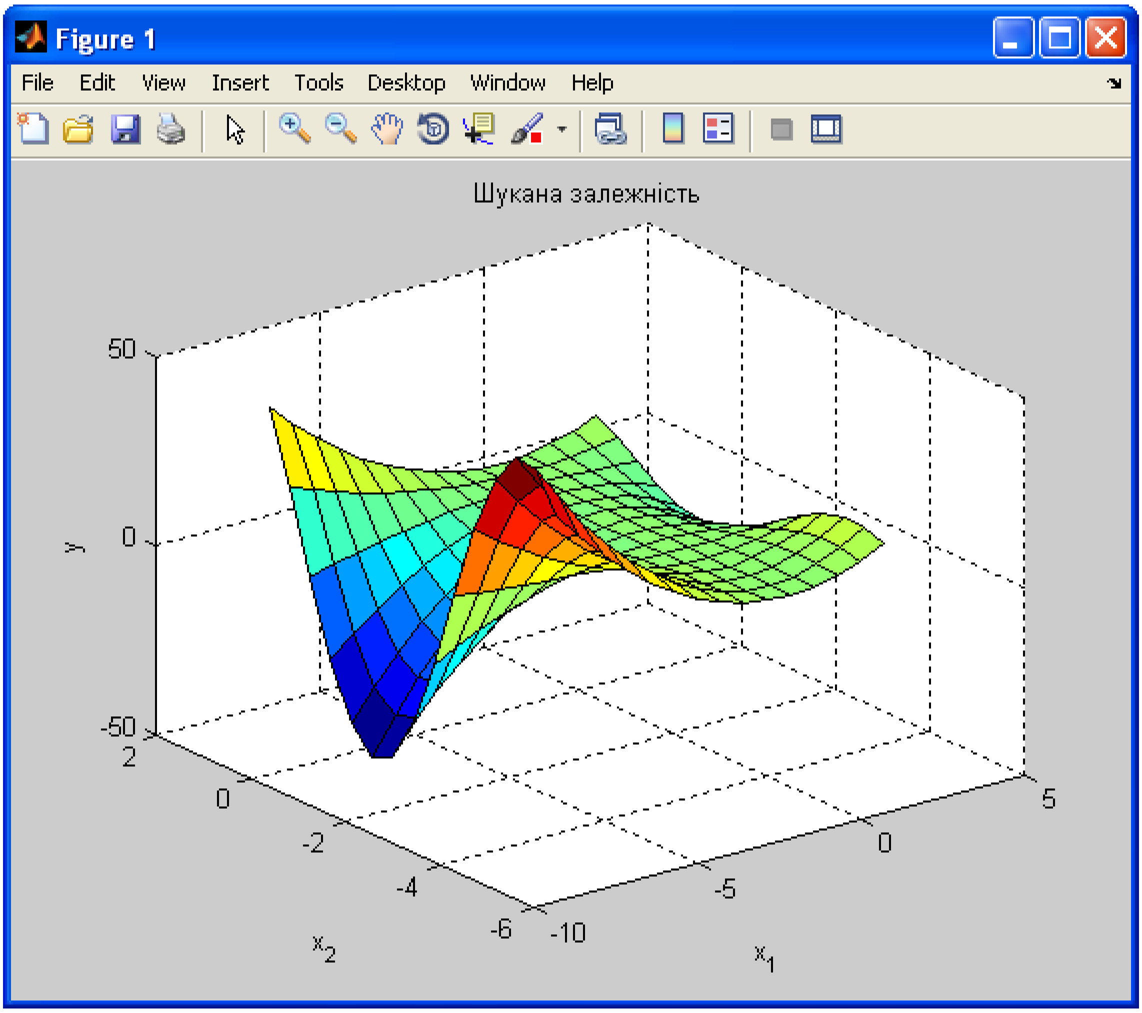


Рис.9.1 – Поверхня «входи-вихід»

Трьохмірне зображення залежності можна побудувати за допомогою такої програми:

% Побудова графіка функції y = xl^2*sin(x2-1)

% в області x1 Î [-7,3] і x2 Î [-4.4,1.7]

```
n = 15; % кількість точок дискретизації
x1 = linspace(-7, 3, n);
x2 = linspace(-4.4, 1.7, n);
y = zeros(n, n);
for j = 1:n
y(j,:) = x1.^2*sin(x2(j)-1);
end
surf(x1, x2, y)
xlabel('x_1'); ylabel('x_2'); zlabel('y'); title('Шукана залежність')
```

Таблиця 9.1 – Навчальна вибірка

x1	*x2*	*y*
-6,5	*-4,0*	*40,51*
-6,3	*-1,0*	*-36.09*
-4,0	*1,5*	*7,67*
2,9	*1,5*	*7,00*
-5,0	*0*	*-21,04*
-2,0	*-3,0*	*3,02*
3,0	*-1,5*	*-539*
2,9	*1,0*	*0*
-5,8	*-1,8*	*-11,27*
-5,0	*-1,0*	*-22,73*
0,5	*1,6*	*0,14*
4,5	*1,0*	*0*
-7,0	*1,1*	*4,90*
-6,0	*0,5*	*-17,26*
5,0	*-0,1*	*-22,28*
-1,5	*0,7*	*-0,66*
-0,9	*-4,0*	*0,78*
-5,7	*-4,4*	*25,11*

-5,2	*-1,6*	*-13,94*
-6,5	*-4,0*	*40,51*

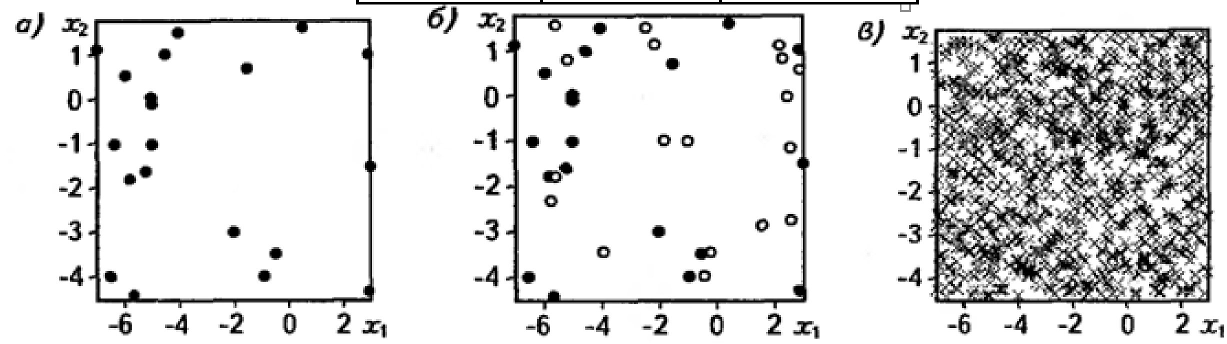


Рис.9.2 – Розподіл даних навчальної та тестової вибірок

а – навчальна вибірка з 20 точок; б – навчальна вибірка з 40 точок;

в – тестова вибірка

Необхідно ідентифікувати залежність нечіткою базою знань Мамдані та порівняти отриману модель з еталонною залежністю в області x_1Î[-7; 3] і x_2Î[-4,4; 1,7]. Адекватність нечіткої моделі необхідно перевірити за критерієм (9.2) на тестовій виборці з 1000 випадково згенерованих пар «входи-вихід».

Входи та вихід нечіткої моделі розглядатимемо як лінгвістичні змінні, значення яких визначаються з наступних терм-множин: {«низький», «середній», «високий»} для x_1 і x_2, і {«низький», «нижче середнього», «середній», «вище середнього», «високий»} для y. Терми представимо нечіткими множинами з гаусівськими функціями приналежності. Вибір гаусівської функції приналежності обумовлений її достатньою гнучкістю і простотою – вона задається лише двома параметрами. Це скорочує розмірність завдання оптимізації при налаштуванні нечіткої бази знань. Графіки вихідних функцій приналежності приведені на рис.9.3а.

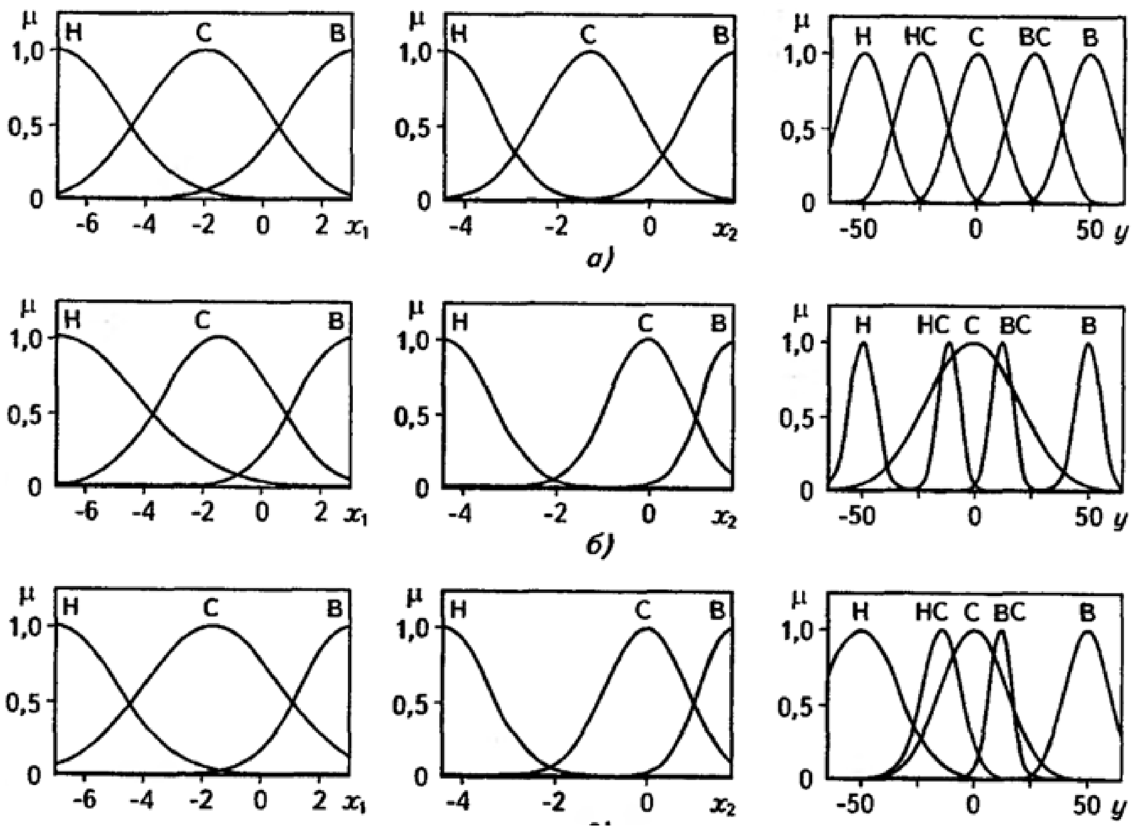


Рис.9.3 – Функції приналежності нечіткої моделі Мамдані

Н – низький; НС – нижче середнього; С – середній;

ВС – вище середнього; В – високий

а – до навчання; б – після навчання на вибірці з 20 точок;

в – після навчання на вибірці з 40 точок

Нечітка база знань згенерована експертом візуально на основі рис.9.1. Вона складається з семи правил, які зведені в табл.9.2.

Таблиця 9.2 – Нечітка база знань Мамдані

x_1	x_2	y	Веса правил (W)		
			до настройки	после настройки на выборке из 20 точек	после настройки на выборке из 40 точек
Низкий	Низкий	Высокий	1	1	1
Низкий	Средний	Низкий	1	1	1
Низкий	Высокий	Высокий	1	1	1
Средний	–	Средний	1	1	1
Высокий	Низкий	Выше среднего	1	0,921	0,977
Высокий	Средний	Ниже среднего	1	0,655	0,479
Высокий	Высокий	Выше среднего	1	0,657	0,105

Як t-норму виберемо максимум. Дефаззіфікацію проводитимемо по методу центру тяжіння, оскільки він забезпечує найкращі показники точності і швидкості настройки нечіткої бази знань.

Поверхня «входи - вихід» вихідної нечіткої моделі показана на рис.9.4а.

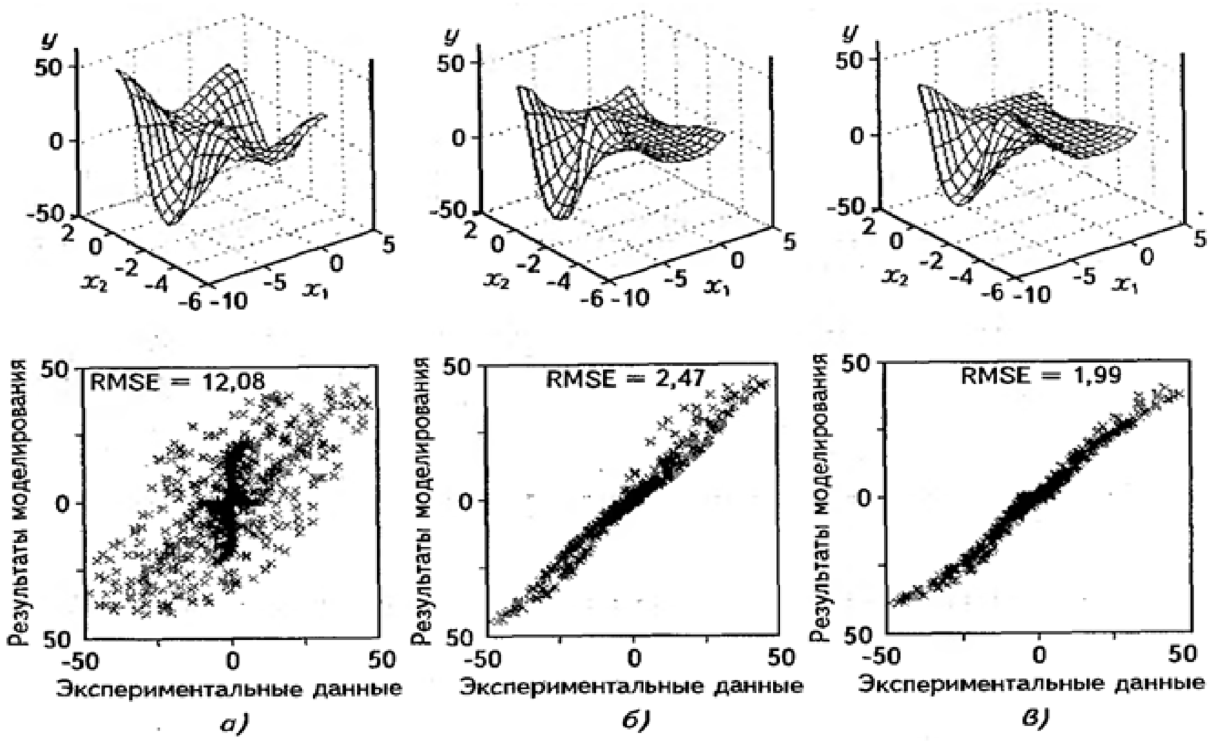


Рис.9.4 – Тестування нечітких моделей Мамдані

а – до навчання; б – після навчання на вибірці з 20 точок;

в – після навчання на вибірці з 40 точок

Як видно з цього рисунка, до настройки нечітка модель відображає основні особливості залежності, що ідентифікується. Тестування моделі (рис.9.4а) показує, що нев'язка між експериментальними даними і результатами нечіткого моделювання чимала – 12,08.

Необхідно настроїти наступні 19 параметрів нечіткої бази знань:

• вагові коефіцієнти 5,6 і 7-го правил;

• коефіцієнти концентрації функцій приналежності термів змінних x_1, x_2 і y;

• координати максимумів функцій приналежності термів «середній» вхідних змінних;

• координати максимумів функцій приналежності не крайніх термів «нижче середнього», «середній» і «вище середнього» вихідної змінної.

Координати максимумів функцій приналежності крайніх термів «низький» і «високий» не настроюються, тому що вони збігаються з межами діапазонів зміни змінних. Не змінюватимемо і вагові коефіцієнти перших чотирьох правил бази знань, оскільки їх істинність не викликає сумнівів.

Настройка моделей типа Мамдані в пакеті Fuzzy Logic Toolbox не передбачена. Для вирішення цього завдання можна використовувати функції нелінійної оптимізації з пакету Optimization Toolbox.

Настройку нечіткої моделі виконаємо квазіньютоновським методом Бройдена-Флетчера-Голфарбда-Шенно на протязі 10 ітерацій.

Знайдені в результаті настройки оптимальні функції приналежності нечітких термів і ваги правил показані на рис.9.3б і в таблиці.9.2, відповідно. Поверхня «входи - вихід» налагодженої нечіткої моделі показана на рис.9.4б. Як видно з цього рисунка, після настройки нечітка модель добре відображає поведінку залежності, що ідентифікується. На це також вказує і мале значення нев'язки на тестовій вибірці, що дорівнює 2,47. Звернемо увагу на те, що навчальна вибірка містить всього 20 пар «входи – вихід», що всього на одиницю більше кількості параметрів, що настроюються. Не дивлячись на це, результати настройки добрі, що пояснюється якісною базою знань. Фактично, вихідна нечітка база знань (див.табл.9.2) вже є грубою моделлю залежності, що ідентифікується, яка на етапі настройки лише довчається.

Проведемо дослідження, як зміниться якість настройки при збільшенні навчальної вибірки. Додамо в навчальну вибірку 20 пар «входи – вихід». Розподіл даних з навчальної вибірки показаний на рис.9.2б: точками вказані дані з табл.2.1, а світлими кружками – нові дані. Функції приналежності нечітких термів і ваги правил після 15 ітерацій навчання показані на рис.9.3в і в табл.9.2, відповідно. Збільшення навчальної вибірки забезпечило зниження помилки тестування до значення 1,99 (див.рис.2.4в).

На рис.9.5 показані криві навчання нечіткої моделі. Вони відображають залежності помилок ідентифікації (9.2) на навчальній і тестовій вибірках від обсягу (М) навчальної вибірки.

Рис.9.5 – Криві навчання нечіткої моделі Мамдані

Кожна точка кривих навчання розраховувалася як середнє значення результатів експериментів для 10 різних навчальних вибірок. Для дослідження процесу настройки нечіткої моделі використовувалися навчальні вибірки з 20,30, ...,110 пар «входи – вихід». У цих вибірках значення входів генерувалися випадково, а вихід розраховувався по формулі еталонної залежності. Як видно з рис.9.5, із збільшенням обсягу навчальної вибірки зменшується нев’язка на тестовій вибірці, а також скорочується різниця між нев’язками на навчальній і тестовій вибірках.

Настройка нечіткої бази знань Суджено

Введемо наступні позначення:

• P – вектор параметрів функцій приналежності термів вхідних і вихідної змінних;

• B – вектор коефіцієнтів лінійних функцій в висновках правил бази знань Суджено;

• $F(P, B, X_r)$ – результат виведення по нечіткій базі знань Мамдані з параметрами (P, B) при значенні входів X_r.

Згідно з методом найменших квадратів, настройка нечіткої бази знань Суджено зводиться до наступної задачі математичного програмування:

знайти такий вектор (P, B), щоб

$$\text{RMSE} = \sqrt{\frac{1}{M}\sum_{r=1,M}(y_r - F(P,B,X_r))^2} \to \min.$$

На практиці цю задачу вирішують як стандартними алгоритмами оптимізації, наприклад, методом Левенберга-Марквардта, так і спеціально розробленими швидкими алгоритмами на основі фільтру Калмана і методу зворотного поширення помилки. Алгоритм на основі фільтру Кальмана оптимізує лише лінійні параметри нечіткої бази знань – коефіцієнти у висновках правил. ANFIS-алгоритм, що є комбінацією методів найшвидшого спуску і зворотного поширення помилки, оптимізує як лінійні, так і нелінійні параметри нечіткої бази знань. При цьому трикутні норми повинні задаватися функціями, що диференціюються.

Приклад 9.2. Ідентифікувати нелінійну залежність з прикладу 9.1 нечіткою базою знань Суджено. Порівняти результати ідентифікації нечіткими базами знань Мамдані та Суджено.

Для лінгвістичної оцінки вхідних змінних використовуватимемо по два нечіткі терми з гаусівською функцією приналежності. Графіки вихідних функцій приналежності представлені на рис.9.6. Вихідна нечітка база знань з чотирьох правил приведена в табл.9.4.

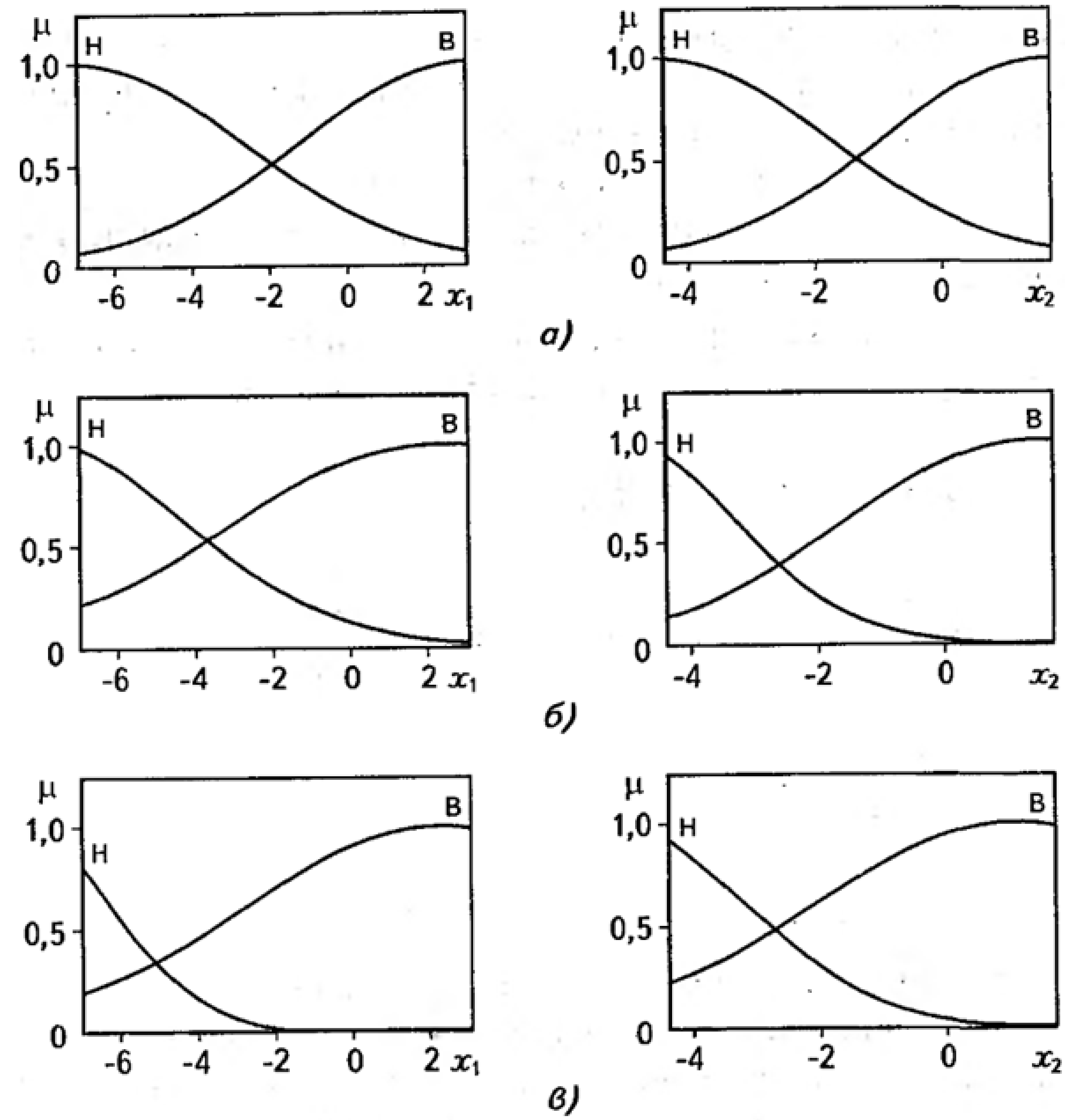


Рис.9.6 – Функції приналежності нечіткої моделі Суджено

Н – низький; В – високий;

а – до навчання; б – після навчання на вибірці з 20 точок;

в – після навчання на вибірці з 40 точок

Таблиця 9.4 – Нечітка база знань Суджено

x_1	x_2	y		
		исходная	после настройки на выборке из 20 точек	после настройки на выборке из 40 точек
Низкий	Низкий	$0+0x_1+0x_2$	$460{,}8+1{,}43x_1+85{,}15x_2$	$642{,}3+8{,}85x_1+103{,}79x_2$
Низкий	Высокий	$0+0x_1+0x_2$	$-47{,}51-0{,}33x_1+40{,}22x_2$	$-82{,}62-2{,}2x_1+58{,}8x_2$
Высокий	Низкий	$0+0x_1+0x_2$	$-93{,}11+3{,}89x_1-20{,}17x_2$	$25{,}81+2{,}13x_1+4{,}62x_2$
Высокий	Высокий	$0+0x_1+0x_2$	$4{,}97-0{,}9x_1+3{,}21x_2$	$-3{,}61-0{,}73x_1+3{,}25x_2$

По графіку залежності (рис.9.1), що ідентифікується, важко оцінити коефіцієнти у висновках правил, тому у вихідній базі знань вони прийняті рівними нулю. При будь-яких значеннях входів нечітка модель видасть нуль. Поверхня «входи – вихід» вихідної нечіткої моделі приведена на рис.9.7а.

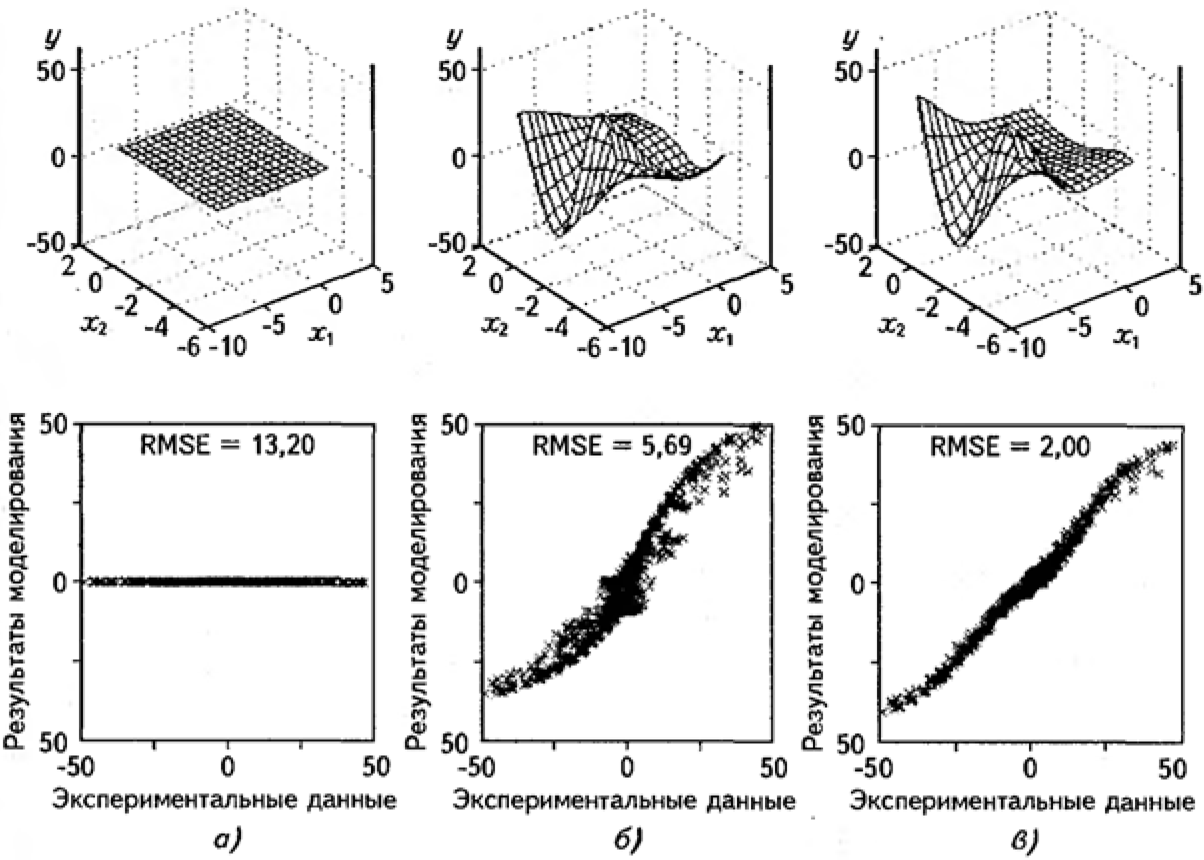


Рис.9.7 – Тестування нечітких моделей Суджено

а – до навчання; б – після навчання на вибірці з 20 точок;

в – після навчання на вибірці з 40 точок

Настроїмо 20 параметрів нечіткої бази знань: по три коефіцієнти у висновках кожного з чотирьох нечітких правил і по два параметри функцій приналежності для кожного з чотирьох термів вхідних змінних. Після 100 ітерацій ANFIS-алгоритму помилка тестування зменшилася з 13,2 до 5,69. На рис.9.7б показана поверхня «входи – вихід», що відповідає нечіткій базі знань Суджено, налагодженій на вибірці з 20 точок. Як видно з рисунка, налагоджена нечітка модель відображає лише основні особливості залежності, що ідентифікується. Низька якість ідентифікації пов’язана з малим обсягом навчальної вибірки – кількість пар «входи - вихід» дорівнює числу параметрів, що настроюються. Якість ідентифікації можна істотно поліпшити, збільшуючи навчальну вибірку. Результати настройки нечіткої бази знань Суджено по навчальній вибірці з 40 пар «входи – вихід» приведені на рис.9.7в.

На рис.9.8 показані криві навчання нечіткої моделі Суджено.

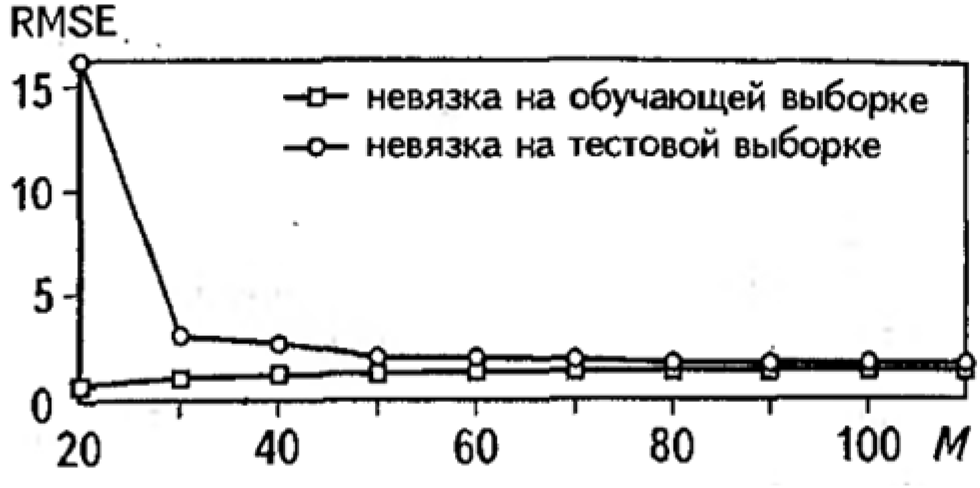


Рис.9.8 – Криві навчання нечіткої моделі Суджено

Кожна точка кривих навчання розраховувалась як середнє значення результатів експериментів для 10 різних навчальних вибірок. Навчання проводилося впродовж 100 ітерацій ANFIS-алгоритму. Як видно з рис.9.8, обсяг навчальної вибірки істотно впливає на якість ідентифікації нечіткою моделлю Суджено.

Для порівняння результатів ідентифікації нечіткими базами знань Мамдані і Суджено на рис.9.9 приведені криві навчання при різній тривалості навчання.

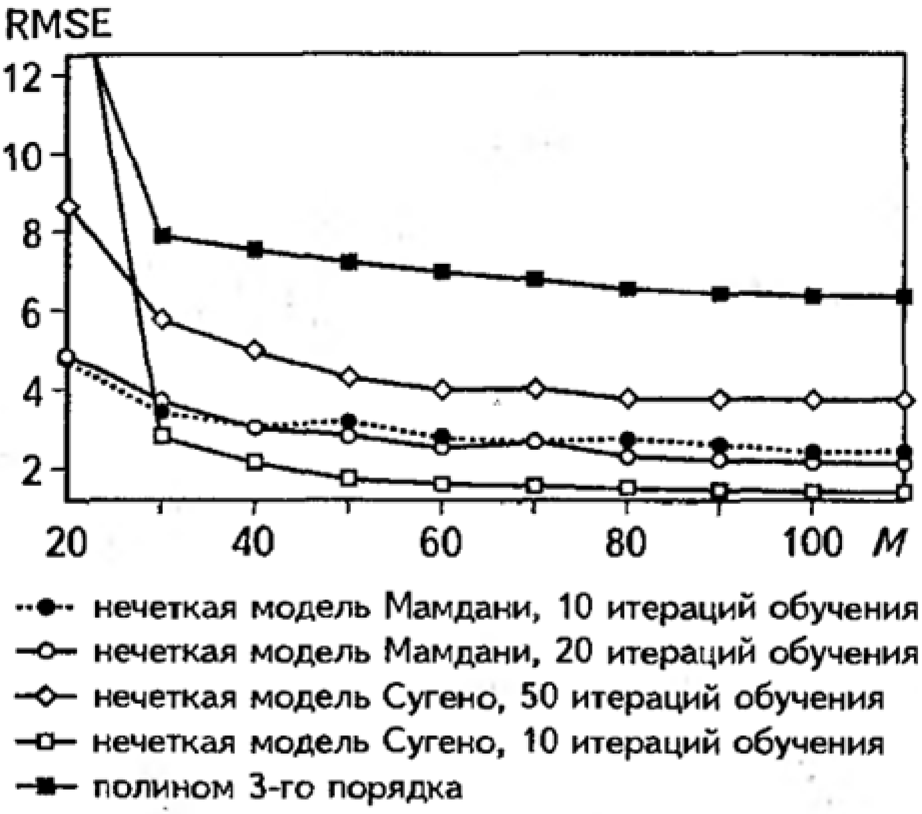


Рис.9.9 – Криві навчання для нечітких баз знань Мамдані та Суджено

Для прикладу приведена також крива навчання при апроксимації поліномом третього ступеня. При малих навчальних вибірках якість ідентифікації істотно вище для нечіткої бази знань Мамдані. Це пояснюється тим, що початкова, заснована на експертних висловлюваннях, нечітка модель Мамдані вже відображає основні особливості залежності, що ідентифікується. При малих навчальних вибірках настройка не повинна бути тривалою, аби уникнути перенавчання. Із збільшенням навчальної вибірки кращу якість ідентифікації забезпечує база знань Суджено. Добрі результати виходять, коли обсяг навчальної вибірки в 2-3 рази перевищує кількість параметрів, що настроюються. При великих навчальних вибірках настає ефект насичення – точність ідентифікації практично не покращується із збільшенням числа спостережень.

Звернемо увагу, що після налаштування база знань Мамдані залишається прозорою, її параметри – функції приналежності – легко інтерпретуються лінгвістичними термами. Для баз знань Суджено типове явище – складність змістовної інтерпретації її параметрів. Як приклад вкажемо на висновки правил в табл.9.4, а також на функцію приналежності терму «низький» змінної x_1 на рис.9.7в. Тому для завдань, де важливіша точність ідентифікації, доцільним буде використання нечітких баз знань Суджено, а для завдань, де важливішим є пояснення і обґрунтування прийнятого рішення, нечіткі бази знань Мамдані матимуть перевагу.

Настройка нечіткої бази знань для задач класифікації

Настройка нечіткого класифікатора є знаходженням таких параметрів функцій приналежностей термів вхідних змінних і вагових коефіцієнтів правил, які мінімізують відхилення між бажаною і дійсною поведінкою нечіткою класифікатора на навчальній вибірці. Критерій близькості можна визначити різними способами.

Перший спосіб полягає у виборі в якості критерія настройки відсотка помилок класифікації на навчальній вибірці. Введемо наступні позначення:

• P – вектор параметрів функцій приналежності термів вхідних і вихідної змінних;

• W – вектор вагових коефіцієнтів правил бази знань;

• $F(P, W, X_r)$ – результат виведення по нечіткій базі знань Мамдані з параметрами (P, W) при значенні входів X_r.

Настройка нечіткого класифікатора зводиться до наступної задачі оптимізації: *знайти такой вектор (P, W), щоб*

$$\frac{100\%}{M}\sum_{r=\overline{1,M}}\Delta_r \to \min,$$

де D_r *– помилка класифікації об'єкта* X_r *:*

$$\Delta_r = \begin{cases} 1, & \text{если } y_r \neq F(X_r, P, W), \\ 0, & \text{если } y_r = F(X_r, P, W). \end{cases}$$

Переваги критерію настройки (10.3) полягають в його простоті і ясній змістовній інтерпретації. Відсоток помилок широко використовується як критерій навчання різних систем розпізнавання образів. Цільова функція задачі оптимізації (10.3) набуває дискретних значень. Це ускладнює використання градієнтних методів оптимізації, оскільки на протяжних плато цільової функції алгоритми оптимізації «застряють». Особливо важко підібрати відповідні параметри градієнтних алгоритмів (наприклад, прирости аргументів для розрахунку часткових похідних) при невеликій вибірці даних.

Другий спосіб використовує як критерій настройки відстань між результатом виводу у вигляді нечіткої множини

$$\left(\frac{\mu_{d_1}(X)}{d_1}, \frac{\mu_{d_2}(X)}{d_2}, \ldots, \frac{\mu_{d_m}(X)}{d_m} \right)$$

і значенням вихідний змінної в навчальній вибірці. Для цього вихідну змінну в навчальній вибірці

$$(X_r, y_r), \quad r = \overline{1, M},$$

фаззіфікують таким чином:

$$\left. \begin{array}{ll} \tilde{y} = (1/d_1,\ 0/d_2, \ldots, 0/d_m), & \text{если } y = d_1 \\ \tilde{y} = (0/d_1,\ 1/d_2, \ldots, 0/d_m), & \text{если } y = d_2 \\ \ldots & \\ \tilde{y} = (0/d_1,\ 0/d_2, \ldots, 1/d_m), & \text{если } y = d_m \end{array} \right\}. \qquad (10.4)$$

В цьому випадку настройка нечіткого класифікатора зводиться до наступної задачі оптимізації: знайти такий вектор (P, W), щоб

$$\sqrt{\frac{1}{M}\sum_{r=1}^{M}\sum_{j=1}^{m}\left(\mu_{d_j}(y_r)-\mu_{d_j}(X_r,P,W)\right)^2} \rightarrow \min\ , \tag{10.5}$$

де Mdj – ступінь приналежності значення вихідної змінної y в r-парі навчальної вибірки до рішення d_j у відповідності з (10.4);

Mdj(Xr, P, W)

– ступінь приналежності входа нечіткої моделі з параметрами (P, W) до рішення d_j, що визначається за формулою (10.1) при значеннях входів із r-пари навчальної вибірки X_r.

Цільова функція в задачі (10.5) не має протяжних плато, тому вона придатна до оптимізації градієнтними методами. Проте результати оптимізації інколи незадовільні: нечітка база знань, що мінімізує (10.5), не завжди забезпечує також і мінімум помилок класифікації. Це пояснюється тим, що точки, близькі до меж розділу класів, вносять майже однаковий вклад до критерію настройки як при правильній, так і при помилковій класифікації.

Третій спосіб успадковує достоїнства попередніх способів. Ідея полягає в тому, аби вклад помилково класифікованих об'єктів в критерій настройки збільшувати за допомогою множення відстані

$$\sqrt{\sum_{j=1}^{m}(\mu_{d_j}(y_r)-\mu_{d_j}(X_r,P,W))^2}$$

на штрафний коефіцієнт. В результаті задача оптимізації приймає такий вигляд:

$$\sqrt{\frac{1}{M}\sum_{r=1}^{M}(\Delta_r\cdot\text{penalty}+1)\sum_{j=1}^{m}(\mu_{d_j}(y_r)-\mu_{d_j}(X_r,P,W))^2} \rightarrow \min, \tag{10.6}$$

На керовані змінні P зазвичай накладають обмеження, що забезпечують лінійну впорядкованість елементів терм-множин. Такі обмеження не дозволяють алгоритмам оптимізації зробити, наприклад, нечітку множину «низький» більше «високого». Крім того, ядра нечітких множин не повинні виходити за межі діапазонів зміни відповідних змінних. Такими обмеженнями забезпечується прозорість нечіткої бази знань після настройки, тобто можливість змістовної інтерпретації правил. Що стосується вектора W, то його координати повинні знаходитися в діапазоні [0, 1]. Якщо до рівня інтерпретабельності бази знань пред'являються високі вимоги, то ваги правил не настроюють, залишаючи їх рівними 1. Можливий і проміжний варіант, коли вагові коефіцієнти можуть набувати значень 0 або 1. В цьому випадку нульове значення вагового коефіцієнта еквівалентне виключенню правила з нечіткої бази знань.

Задачі (10.3), (10.5) і (10.6) можуть бути вирішені різними технологіями оптимізації, серед яких часто застосовують метод найскорішого спуску, квазіньютонівські методи і генетичні алгоритми.

Параметри функцій приналежності і ваги правил можна настроювати одночасно або окремо. При настройці лише ваг правил обсяг обчислень можна значно скоротити, оскільки вхідні в (10.1) міри приналежності Mjp(X*i) не залежать від W.

Для цього на початку оптимізації потрібно розрахувати ступені виконання правил при одиничних вагових коефіцієнтах (Wjp)=1 для кожного об'єкта з навчальної вибірки:

$$g_{jp}(X_r) = \min_{i=1,n} \mu_{jp}(x_{ri}), \quad j=\overline{1,m}, \; p=\overline{1,k_j}, \; r=\overline{1,M}.$$

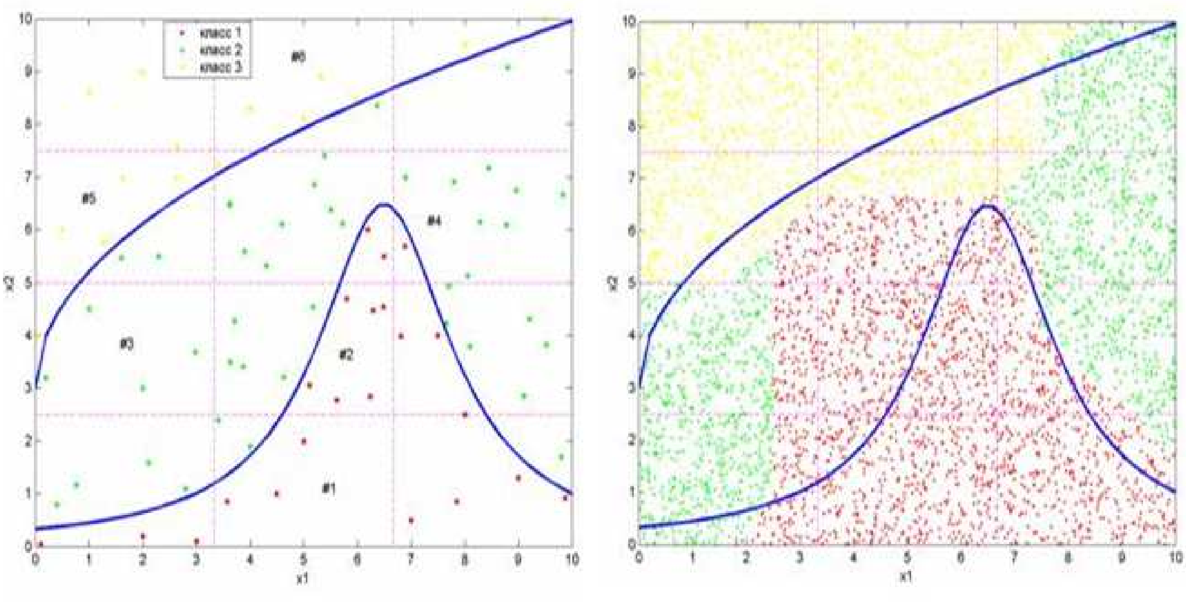
Для нових вагових коефіцієнтів ступені приналежності об'єкта X_r класам d_j перераховуються так:

$$\mu_{d_j}(X_r) = \max_{p=1,k_j} w_{jp} g_{jp}(X_r), \quad j=\overline{1,m}.$$

Приклад 1. Розглядається об'єкт з двома входами x_1, x_2 Î [0, 10] і одним виходом y, який може приймати одне з трьох дискретних значень $\{d_1, d_2, d_3\}$ відповідно до вирішальних правил:

$$y = \begin{cases} d_1, \text{ если } x_2 < \dfrac{14{,}6}{2{,}25 + (x_1 - 6{,}5)^2}, \\ d_2, \text{ если } \dfrac{14{,}6}{2{,}25 + (x_1 - 6{,}5)^2} < x_2 < 2{,}2\sqrt{x_1} + 3, \\ d_3, \text{ если } x_2 > 2{,}2\sqrt{x_1} + 3. \end{cases} \qquad (10.7)$$

Розділяючі криві, а також навчальна і тестова вибірки показані на рис.10.2. Вибірки містять 80 і 5000 пар «входи – вихід» відповідно. У вибірках значення входів вибиралися випадково, а значення виходу розраховувалися по формулі (10.7).



а) б)

Рис.10.2 – Навчальна (а) і тестова (б) вибірки

На основі цих вибірок спроектуємо нечіткий класифікатор. Входи нечіткого класифікатора розглядатимемо як лінгвістичні змінні, значення яких визначаються з наступних терм-множин: {«низький», «середній», «високий»} для x_1 і {«низький», «нижче середнього», «вище середнього», «високий») для x_2 (див.табл.10.2).

По рис.10.2 експерт згенерував шість нечітких правил класифікації, які зведені в табл.10.2. Зони дії правил позначені на рис.10.2 символами #1, #2, …. #6.

Таблиця 10.2 – Нечітка база знань

x_1	x_2	y	Веса правил, W			
			До настройки	Классификатор I	Классификатор II	Классификатор III
Средний	Низкий	d_1	1	0,767	0,751	0,779
Средний	Ниже среднего	d_1	1	0,513	0,395	0,539
Низкий	Ниже среднего	d_2	1	1,000	1,000	1,000
Высокий	Выше среднего	d_2	1	0,570	1,000	0,783
Низкий	Выше среднего	d_3	1	0,816	0,490	0,697
Средний	Высокий	d_3	1	0,020	0,016	1,000

Формалізацію термів здійснимо симетричною гаусівською функцією приналежності (рис.10.3а). До настройки коефіцієнти концентрацій всіх функцій приналежності дорівнюють 2. Координати максимумів вибрані так, щоб розбити інтервал [0, 10] на три (для x_1) і на чотири (для x_2) рівні частини.

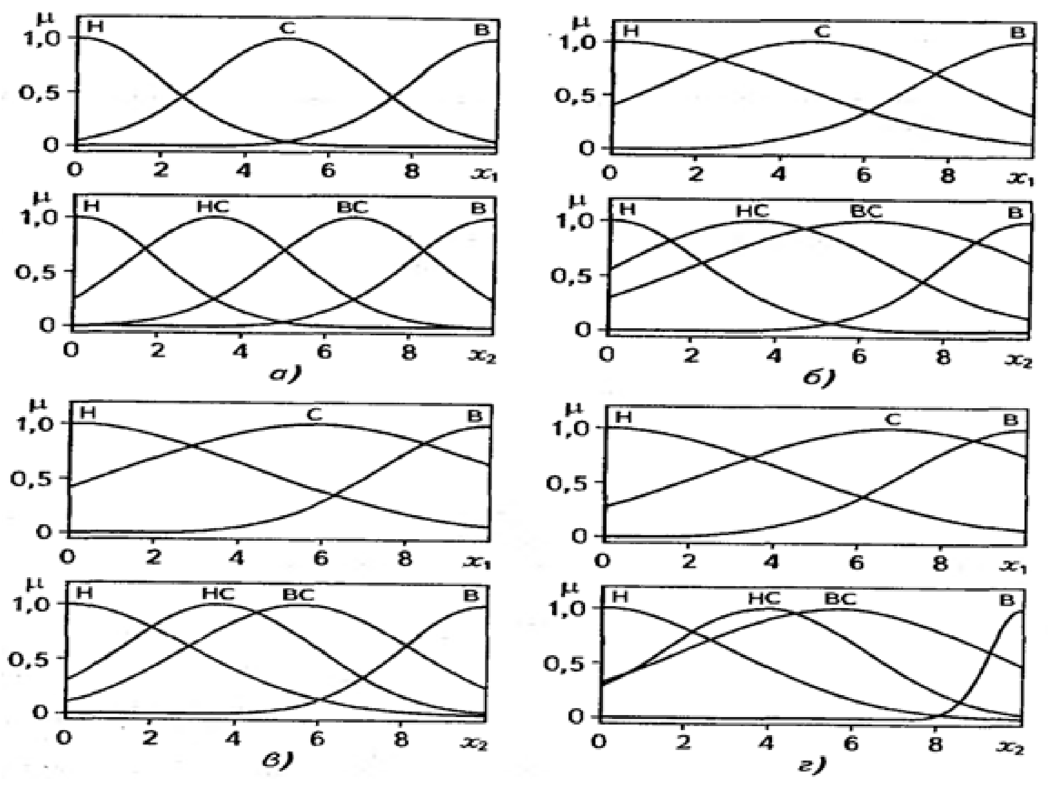


Рис.10.3 – Функції приналежності

Н – низький; НС – нижче середнього; С – середній; ВС – вище середнього; В – високий

а – вихідний класифікатор; б – класифікатор I; в – класифікатор II; г – класифікатор III

Вихідна нечітка база знань з шістьма правилами грубо відображає нелінійні розділяючі криві – на тестовій вибірці помилково класифіковано 26,6% об’єктів (рис.10.4а).

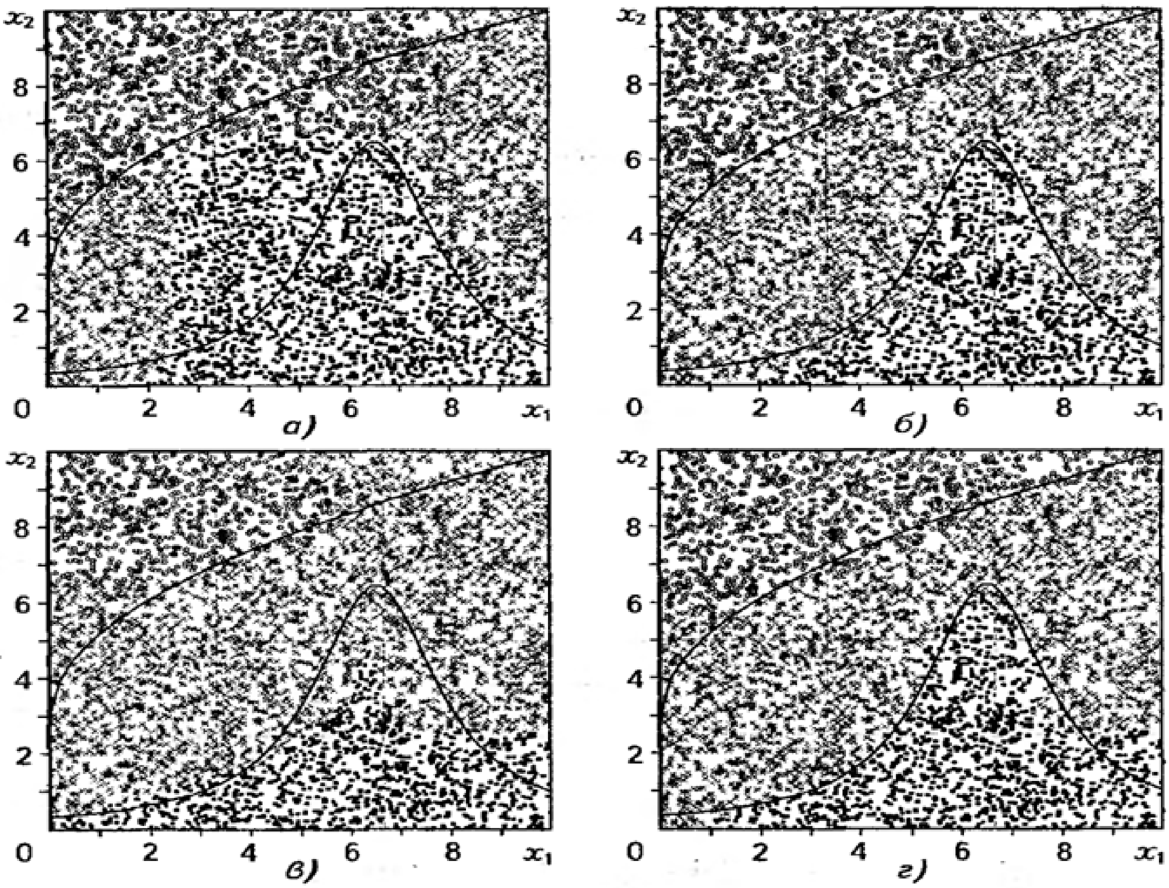


Рис.10.4 – Класифікація на тестовій виборці

а – вихідний класифікатор; б – класифікатор I;

в – класифікатор II; г – класифікатор III

Після настройки ваг правил кількість помилок зменшилася до 15%, проте безпомилковість нечіткого класифікатора залишається на рівні наступного простого дерева рішень:

```
Если ((x1>1.2929) & (x2≤1)),                             то y = d1,
Если ((x1>4.6335) & (x2>1) & (x1≤7.5) & (x2≤6)),         то y = d1,
Если ((x2>8.3607) & (x1>5.3301)),                        то y = d3,
Если ((x1≤5.3301) & (x2>6.9107)),                        то y = d3,
Если ((x1≤1.2929) & (x2>3.4988) & (x2≤6.9107)),          то y = d3,
Иначе,                                                      y = d2.
```

Низька безпомилковість нечіткого класифікатора після настройки ваг правил пояснюється «поганими» функціями приналежності. Тому необхідно модифікувати не лише ваги правил, але і функції приналежності. Налаштуємо наступні 16 параметрів нечіткого класифікатора:

- три координати максимумів функцій приналежностей термів «середній», «нижче середнього» і «вище середнього»;

- сім коефіцієнтів концентрацій функцій приналежностей термів вхідних змінних;

- шість вагових коефіцієнтів правил бази знань.

Результати настройки з використанням різних критеріїв приведені на рис.10.4 і в табл.10.5. Використовувалися такі критерії настройки: критерій I - формула (10.3); критерій II - формула (10.5); критерій III - формула (10.6). Результати тестування класифікаторів приведені в табл.10.6. Класифікатор I налагоджений по критерію I, класифікатор II - по критерію II і класифікатор III - по критерію III. Критерії I і III забезпечують краще налаштування нечіткого класифікатора, проте слід пам'ятати, що підбір відповідних параметрів градієнтних алгоритмів при оптимізації по критерію I часто віднімає багато часу.

Таблиця 10.6 – Результати тестування класифікаторів

Классификатор	Критерий I, %	Критерий II	Критерий III (penalty = 9)	Безошибочность иа тестовой выборке, %
Исходный	33,75	0,658	1,997	26,60
Классификатор I	6,25	0,727	1,013	9,78
Классификатор II	18,75	0,646	1,221	16,42
Классификатор III	8,75	0,676	1,018	9,22
Дерево решений	7,50	–	–	15,24

ТЕМА 9. НЕЧІТКИЙ БАГАТОКРИТЕРАЛЬНИЙ АНАЛІЗ ВАРІАНТІВ

Нечіткі цілі, обмеження та рішення. Схема Белмана-Заде

Задача прийняття рішень в нечітких умовах відрізняється від задачі прийняття рішень в загальному вигляді тим, що один або декілька елементів моделі прийняття рішень задаються нечіткою множиною...

Прийняття рішень за допомогою нечіткої теорії очікуваної корисності

Прийняття рішень за принципом Белмана–Заде передбачає використання функції мети, яка виступає в якості критерію прийняття рішення. Проте на практиці в складних системах досить важко визначити критерій прийняття рішення. В таких ситуаціях використовуються суб'єктивні оцінки корисності альтернатив, отримані від експертів...

Прийняття рішень з використанням нечіткого логічного висновку

Нечіткий логічний висновок являє собою апрокимацію залежносі між входами і виходами системи за допомогою нечіткої бази знань та операцій над нечіткими множинами...

Нечіткий багатокритеріальний аналіз варіантів

Завдання багатокритеріального аналізу полягає у впорядковуванні елементів множини P по критеріях із множини G...

Нечіткий багатокритеріальний аналіз бренд-проектів

Як приклад прийняття рішень в нечітких умовах за схемою Беллмана-Заде розглянемо порівняння чотирьох проектів ...

Аналіз варіантів «що буде, якщо …»

При багатокритеріальному аналізі часто виникає питання: «Що необхідно змінити в деякій альтернативі, аби вона стала найкращою?» Для відповіді на нього треба знати, наскільки чутливо прийняте рішення до експертних парних порівнянь...

Нечіткі цілі, обмеження та рішення

Задача прийняття рішень в нечітких умовах відрізняється від задачі прийняття рішень в загальному вигляді тим, що один або декілька елементів моделі прийняття рішень задаються нечіткою множиною. Необхідно вибрати рішення, яке б задовольняло нечітким обмеженням

$$\tilde{C} \subseteq D,$$

заданим на множині рішень *D*, та максимізувало критерій ефективності

$$\tilde{G} \subseteq Y,$$

заданий на множині станів системи *Y*. Залежність стану системи від прийнятого рішення описується нечітким відображенням

$$\tilde{\varphi} = D \rightarrow Y$$

В залежності від складових моделі прийняття рішення, які визначаються з певним ступенем невизначеності, можна виділити такі підходи прийняття рішень в нечітких умовах:

· _прийняття рішень за принципом Белмана–Заде;

· прийняття рішень за допомогою нечіткої теорії очікуваної корисності;

· прийняття рішень за допомогою нечіткого логічного висновку (задачі ситуативного управління).

Прийняття рішень в нечітких умовах за схемою Белмана-Заде

Традиційний підхід до прийняття рішення на основі нечіткої логіки базується на принципі Белмана–Заде, який розглядає нечітке рішення як об'єднання нечітких цілей та обмежень.
У 1970р. Белман і Заде опублікували статтю «Decision – Making in Fuzzy Environment», де розглянули процес ухвалення рішень в умовах невизначеності, коли цілі і обмеження задані нечіткими множинами. Ухвалення рішення – це вибір альтернативи, яка одночасно задовольняє і нечітким цілям, і нечітким обмеженням. У цьому сенсі цілі та обмеження є

*симетричними відносно рішення. Це стирає відмінності між ними і дозволяє представити рішення як злиття нечітких цілей і обмежень. Нехай X={x} – множина альтернатив. Нечітку мету G будемо ототожнювати з нечіткою множиною G в X. Наприклад, якщо альтернативами є дійсні числа, то X=**R** і нечітка мета формулюється як*

«*x* повинно бути біля 10». Її можна представити нечіткою множиною з такою функцією приналежності:

$$\mu_G(x) = \frac{1}{1+(x-10)^2}, x \in X. \tag{1}$$

Аналогічним чином нечітке обмеження C визначається як деяка нечітка множина на універсальній множині X. Наприклад, нечітке обмеження «*x* повинно бути значно більше 8» при *X*=**R** можна представити нечіткою множиною з такою функцією приналежності

$$\mu_C(x) = \begin{cases} 0, & \text{если } x < 5, \\ \dfrac{1}{1+\exp(-0{,}8(x-8))}, & \text{если } x \geq 5. \end{cases} \tag{2}$$

Нечітке рішення D також визначається як нечітка множина на універсальній множині альтернатив *X*. Функція приналежності цієї нечіткої множини показує, наскільки добре рішення задовольняє *нечітким цілям* ТА *нечітким обмеженням.* Логічній операції ТА, яка пов'язує цілі з обмеженнями, відповідає перетинання нечітких множин. Отже, рішення – це перетинання нечіткої мети з нечітким обмеженням:

$$\tilde{D} = \tilde{G} \cap \tilde{C} \tag{3}$$

Якщо на множині альтернатив D = $\{d_1, d_2, ...\}$ задана нечітка функція мети G та нечітке обмеження (відображення є тотожним), то нечітким рішенням цієї задачі виступає множина, утворена в результаті перетину нечіткої мети та обмеження (рис. 1).

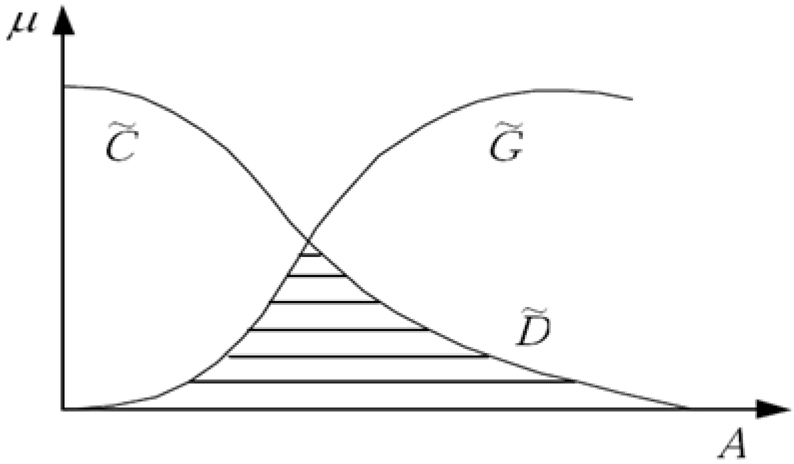


Рис.1 – Знаходження нечіткого рішення як перетину нечітких цілей і нечітких обмежень

Приклад 11.1. Нечітка мета G і нечітке обмеження C сформульовані так: G*:* «*x* повинно бути біля 10» і C : «*x* має бути значно більше 8». Функції приналежності нечітких множин G і C задані виразами (1) і (2). Нечітке рішення D знайдемо по формулі (3). Враховуючи, що перетину нечітких множин відповідає операція мінімуму над функцією приналежності, отримуємо:

$$\mu_D(x) = \begin{cases} 0, & \text{если } x < 5, \\ \min\left(\dfrac{1}{1+\exp(-0{,}8(x-8))}, \dfrac{1}{1+(x-10)^2}\right), & \text{если } x \geq 5. \end{cases}$$

Взаємозв'язок між нечіткими метою, обмеженням і рішенням показаний на рис.2. Мета і обмеження конфліктують між собою, тому в нечіткій множині D немає жодного елементу з мірою приналежності, що дорівнює 1.Значить, не існує альтернативи, яка повністю задовольняє і мету, і обмеження. Як чітке рішення в таких випадках зазвичай обирають альтернативу з максимальною мірою приналежності нечіткій множині. При прийнятті рішень за схемою Белмана-Заде не робиться жодної відмінності між метою і обмеженнями. Всяке розділення на мету і обмеження є умовним:

у формулі (3) можна поміняти місцями мету з обмеженням, при цьому рішення не зміниться. У традиційній теорії прийняття рішень подібні заміни функції переваги на обмеження недопустимі.

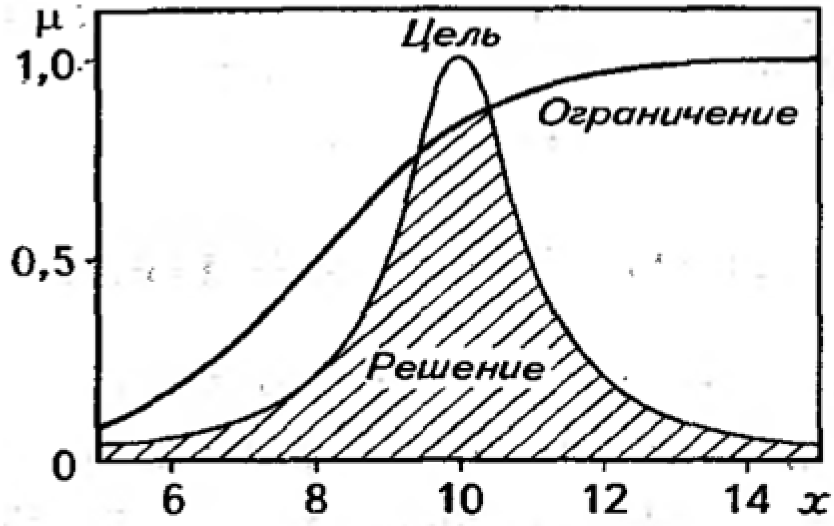


Рис.11.2 – Прийняття рішення за схемою Белмана-Заде

У загальному випадку, коли є *n* цілей і *m* обмежень, результуюче рішення за схемою Беллмана-Раде визначається перетинанням всіх цілей і обмежень:

$$\tilde{D} = \tilde{G}_1 \cap \tilde{G}_2 \cap \ldots \cap \tilde{G}_n \cap \tilde{C}_1 \cap \tilde{C}_1 \cap \ldots \cap \tilde{C}_m$$

і відповідно

$$\mu_D = \mu_{G_1} \wedge \mu_{G_2} \wedge \ldots \wedge \mu_{G_n} \wedge \mu_{C_1} \wedge \mu_{C_2} \wedge \ldots \wedge \mu_{C_m}$$

Оптимальною вважається альтернатива *d* є *D*, яка має максимальний є ступінь належності нечіткому рішенню

$$\max_{d \in D} \mu_D(d)$$

Якщо відображення j не є тотожним, то нечітке рішення знаходиться як перетин нечітких обмежень C і прообразу g , який є результатом відображення мети G на множину альтернатив.

До цих пір передбачалося, що всі цілі та обмеження, що входять в D, мають однакову важливість. Більш звична ситуація, в якій задоволення одних цілей і (або) обмежень важливіше, ніж інших. Позначимо через a_iє(0, 1) – коефіцієнт відносної важливості i-й мети, а через b_iє(0, 1) – коефіцієнт відносної важливості j-го обмеження:

$$\sum_{i=\overline{1,n}} \alpha_i + \sum_{j=\overline{1,m}} \beta_j = 1.$$

Тоді функція приналежності рішення визначається так:

$$\mu_D = (\mu_{G_1})^{\alpha_1} \wedge (\mu_{G_2})^{\alpha_2} \wedge \ldots \wedge (\mu_{G_n})^{\alpha_n} \wedge (\mu_{C_1})^{\beta_1} \wedge (\mu_{C_2})^{\beta_2} \wedge \ldots \wedge (\mu_{C_m})^{\beta_m}.$$

Чим менше коефіцієнт відносної важливості, тим відповідна нечітка множина мети або обмеження стає більш розмазаною та його роль у прийнятті рішення знижується. На рис. 3 показані нечіткі рішення при різних коефіцієнтах важливості мети і обмеження з прикладу 1.

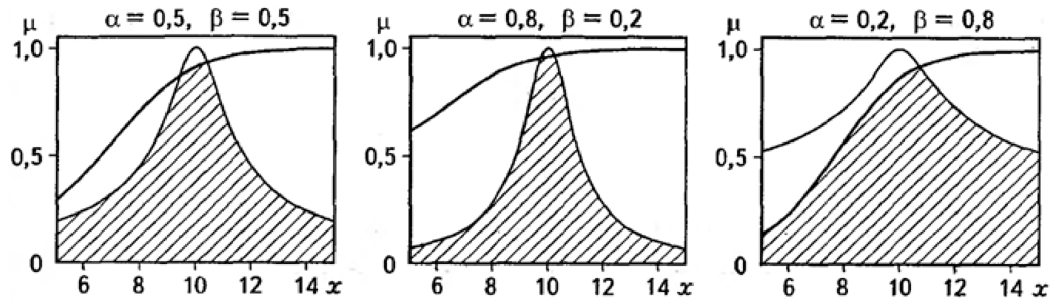


Рис.3 – Прийняття рішень при різній важливості мети та обмеження

Прийняття рішень за допомогою нечіткої теорії очікуваної корисності

Прийняття рішень за принципом Белмана–Заде передбачає використання функції мети, яка виступає в якості критерію прийняття рішення. Проте на практиці в складних системах досить важко визначити критерій прийняття рішення. В таких ситуаціях використовуються суб'єктивні оцінки корисності альтернатив, отримані від експертів.

Один із підходів прийняття рішень в задачах полягає у порівнянні альтернатив на основі деяких відношень. Головною перевагою цього підходу є можливість пошуку оптимального рішення навіть за відсутності даних про результат порівняння деяких альтернатив. Це досягається за рахунок використання нечітких відношень, що характеризує ступінь виконання операцій над альтернативами. Ступінь належності нечітких відношень описується функціями:

· нечітке відношення байдужості

$$\mu_R(d_1,d_2)=\max[1-\max\{\mu_R(d_1,d_2);\ \mu_R(d_2,d_1)\};\ \min\{\mu_R(d_1,d_2);\ \mu_R(d_2,d_1)\}]$$

· нечітке відношення еквівалентності:

$$\mu_R(d_1,d_2)=\min(\mu_R(d_1,d_2);\ \mu_R(d_2,d_1));$$

· нечітке відношення строгої переваги:

$$\mu_R(d_1,d_2)=\begin{cases}\mu_R(d_1,d_2)-\mu_R(d_2,d_1),\ \textit{якщо}\ \mu_R(d_1,d_2)\ge\mu_R(d_2,d_1)\\ 0,\ \textit{в іншому випадку}.\end{cases}$$

Використання цих відношень дозволяє знайти найкращу альтернативу або підмножину оптимальних альтернатив.

Прийняття рішень з використанням нечіткого логічного висновку

Нечіткий логічний висновок являє собою апрокимацію залежносі між входами і виходами системи за допомогою нечіткої бази знань та операцій над нечіткими множинами. Відображенням *j* множини станів *Y* у множину рішень *D* виступає база знань, яка складається з набору правил. Оптимальність прийнятого в такий спосіб рішення залежить від точності функцій належності величин та бази знань. В більшості випадків прийнятна точність рішень досягається з рахунок настроювання параметрів функцій та вагових коефіцієнтів правил на основі вибірки експериментальних даних.

Система нечіткого логічного висновку, показана на рис.1.

Рис.1 – Система нечіткого логічного висновку

Процес фазифікації полягає у перетворенні вхідної величини у нечітку форму шляхом визначення ступеня належності значення вхідної величини її термам. Слід зауважити, що процес фазифікації передбачає попередній збір експертної інформації та використання процедур її обробки для побудови функцій належності вхідних величин.

Розроблено кілька алгоритмів нечіткого логічного висновку, які переважно відрізняються правилами висновку та здійсненням логічних операцій. На сьогоднішній день найбільш поширеними є моделі нечіткого логічного висновку Сугено і Мамдані.

Умова правила характеризує належність вхідної величини її термам. Висновок визначає значення вихідної величини, причому це значення може бути чітким, нечітким або деяким класом.

В результаті досліджень точності процесу прийняття рішень на основі нечіткої інформації було встановлено зниження точності результату при кількості вхідних змінних більше 7 ±2 [102 . Тому в моделюванні складних систем все частіше використовується ієрархічна система нечіткого логічного висновку (рис.2).

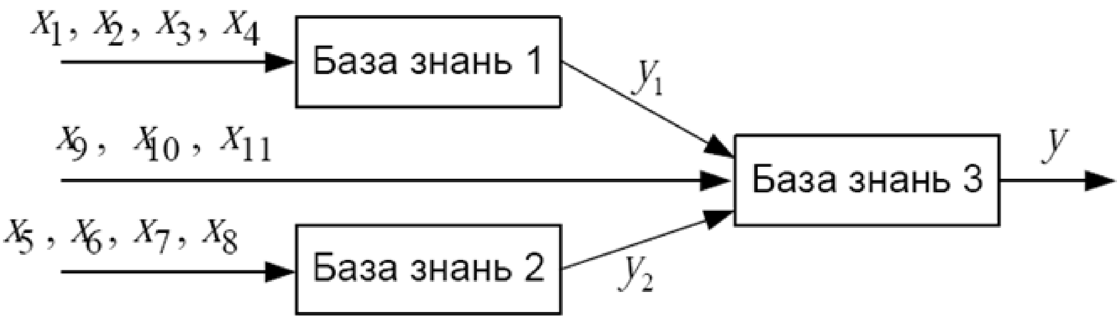


Рис.2 – Ієрархічна система нечіткого логічного висновку

Наведена система дозволяв встановити багатофакторну залежність, використовуючи результати логічного висновку з нечітких баз знань різних рівнів.

Побудова і настроювання нечеткої бази знань є однією з головних задач при розробці нечіткої системи.

Якшо вихідна величина описується нечіткою множиною, то останній етап прийняття рішення в нечіткій системі полягае у дефазифікації вихідної величини. Процедура дефазифікації передбачає перетворення вихідної величини у число. Існують різні методи дефазифікації, проте найбільш поширеним є метод центру тяжіння, який описується формулою

$$y = \frac{\int \mu(y) \cdot y\, dy}{\int \mu(y)\, dy},$$

де $m(y)$ – функція належності нечіткої величини y.

Нечіткий багатокритеріальний аналіз варіантів

Вважаємо відомими:

· $\mathbf{P}=\{P_1, P_2, \ldots, P_k\}$ – множина варіантів, що підлягають багатокритеріальному аналізу;

· $\mathbf{G}=\{G_1, G_2, \ldots, G_k\}$ – множина критеріїв, за якими оцінюють варіанти

Завдання багатокритеріального аналізу полягає у впорядковуванні елементів множини **P** по критеріях із множини **G**.

Нехай $MGi(P_j)$– число в діапазоні [0, 1], яким оцінюється варіант $P_j \in \mathbf{P}$ по критерію $G_i \in \mathbf{G}$: чим більше число MGi(P_j), тим кращий варіант P_j за критерієм G_i , , . Тоді критерій G_i можна представити нечіткою множиною G на універсальній множині варіантів **P:**

$$\tilde{G}_i = \left\{ \frac{\mu_{G_i}(P_1)}{P_1}, \frac{\mu_{G_i}(P_2)}{P_2}, \ldots, \frac{\mu_{G_i}(P_k)}{P_k} \right\}, \tag{1}$$

де $MGi(P_j)$ – міра приналежності елемента P_j нечіткій множині G_i .

Знаходити міри приналежності нечіткої множини (1) зручно методом побудови функцій приналежності на основі парних порівнянь. При використанні цього методу необхідно сформувати матриці парних порівнянь варіантів по кожному критерію. Загальна кількість таких матриць дорівнює кількості критеріїв.

Найкращим варіантом буде той, який одночасно кращий по всіх критеріях. Нечітке рішення D знаходиться як перетинання часткових критеріїв:

$$\tilde{D} = \tilde{G}_1 \cap \tilde{G}_2 \cap \ldots \cap \tilde{G}_n = \left\{ \frac{\min\limits_{i=\overline{1,n}} \mu_{G_i}(P_1)}{P_1}, \frac{\min\limits_{i=\overline{1,n}} \mu_{G_i}(P_2)}{P_2}, \ldots, \frac{\min\limits_{i=\overline{1,n}} \mu_{G_i}(P_k)}{P_k} \right\}. \tag{2}$$

Згідно отриманої нечіткої множини D , найкращим варіантом слід вважати такий, у якого найбільша міра приналежності:

$$D = \arg\ \max(\mu_D(P_1),\ \mu_D(P_2), \ldots,\ \mu_D(P_k)). \tag{3}$$

При нерівноважних критеріях міри приналежності нечіткої множини D знаходять так:

$$\mu_D(P_j) = \min_{i=\overline{1,n}} (\mu_{G_i}(P_j))^{a_i}, \quad j = \overline{1, k}, \tag{4}$$

де a_i – коефіцієнт відносної важливості критерія G_i, $a_1 + a_2 + \ldots + a_n = 1$.

Показник ступеня a_i в формулі (4) концентрує нечітку множину у відповідності з мірою важливості критерія G_i . Коефіцієнти відносної важливості критеріїв можуть бути визначені різними способами, наприклад, з використанням парних порівнянь за шкалою Сааті.

При побудові функцій приналежності по методу парних порівнянь для кожної пари елементів універсальної множини експерт оцінює перевагу одного елементу над іншим по відношенню до властивості нечіткої множини. Такі парні порівняння зручно представляти наступною матрицею:

$$A = \begin{array}{c c} & \begin{array}{cccc} u_1 & u_2 & \dots & u_n \end{array} \\ \begin{array}{c} u_1 \\ u_2 \\ \dots \\ u_n \end{array} & \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & & & \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \end{array},$$

де a_{ij} – рівень переваги елемента u_i над елементом u_j ( ), що визначається за десятибальною шкалою Сааті:

1 – якщо перевага елементу u_i над елементом u_j *відсутня;'*

3 - якщо перевага u_i над u_j *слабка;*

5-якщо перевага u_i над u_j *істотна;*

7 - якщо перевага u_i над u_j *явна;*

9 - якщо перевага u_i над u_j *абсолютна;*

2,4,6,8 – проміжні порівняльні оцінки: 2 – *майже слабка* перевага, 4 – *майже істотна* перевага, 6 – *майже явна* перевага і 8 – *майже абсолютна* перевага.

Матриця попарних порівнянь є діагональною (a_{ij} =1,) та обернено симетричною (a_{ij} =1/ a_{ji},).

Міри приналежностей приймають рівними відповідним координатам власного вектора $W=(w_1, w_2, \dots, w_n)^T$ матриці парних порівнянь A:

m(u_i) = w_i (5)

Власний вектор знаходять з наступної системи рівнянь:

$$\begin{cases} AW = \lambda_{\max} W, \\ w_1 + w_2 + \dots + w_n = 1, \end{cases}$$

де λ_{max} – найбільше власне значення матриці A.

Нечіткий багатокритеріальний аналіз бренд-проектів

Як приклад прийняття рішень в нечітких умовах за схемою Беллмана-Заде розглянемо порівняння чотирьох проектів (P_1 ¸ P_4) виведення бренду на ринок. Для оцінки проектів скористаємося такими критеріями:

G_1 – рівень опрацювання проекту;

G_2 – очікуваний ефект;

G_3 – ризики;

G_4 – швидкість виведення бренду;

G_5 – перспективи розвитку бренду;

G_6 – вартість проекту.

Експертні порівняння проектів по критеріях (G_1 ¸ G_6) приведені в таблиці 11.1. По кожному критерію порівнювалися шість пар проектів.

Таблиця 1 – Парні порівняння проектів за шкалою Сааті

Кри-терий	Экспертные парные сравнения	
G_1	Отсутствует преимущество P_1 над P_2 Слабое преимущество P_1 над P_3 Существенное преимущество P_1 над P_4	Слабое преимущество P_2 над P_3 Существенное преимущество P_2 над P_4 Слабое преимущество P_3 над P_4
G_2	Слабое преимущество P_1 над P_2 Существенное преимущество P_1 над P_3 Явное преимущество P_1 над P_4	Почти слабое преимущество P_2 над P_3 Слабое преимущество P_2 над P_4 Почти слабое преимущество P_3 над P_4
G_3	Существенное преимущество P_1 над P_2 Отсутствует преимущество P_1 над P_3 Явное преимущество P_1 над P_4	Слабое преимущество P_2 над P_4 Существенное преимущество P_3 над P_2 Явное преимущество P_3 над P_4
G_4	Слабое преимущество P_2 над P_1 Отсутствует преимущество P_2 над P_4 Слабое преимущество P_4 над P_1	Существенное преимущество P_3 над P_1 Почти слабое преимущество P_3 над P_2 Слабое преимущество P_3 над P_4
G_5	Слабое преимущество P_2 над P_1 Отсутствует преимущество P_2 над P_3 Слабое преимущество P_3 над P_1	Существенное преимущество P_4 над P_1 Слабое преимущество P_4 над P_2 Почти слабое преимущество P_4 над P_3
G_6	Явное преимущество P_2 над P_1 Слабое преимущество P_2 над P_3 Отсутствует преимущество P_2 над P_4	Слабое преимущество P_3 над P_1 Явное преимущество P_4 над P_1 Слабое преимущество P_4 над P_3

Експертним висловам відповідають наступні матриці парних порівнянь:

$$\mathbf{A}(G_1)=\begin{bmatrix}1 & 1 & 3 & 5\\ 1 & 1 & 3 & 5\\ 1/3 & 1/3 & 1 & 3\\ 1/5 & 1/5 & 1/3 & 1\end{bmatrix};\quad \mathbf{A}(G_2)=\begin{bmatrix}1 & 3 & 5 & 7\\ 1/3 & 1 & 2 & 3\\ 1/5 & 1/2 & 1 & 2\\ 1/7 & 1/3 & 1/2 & 1\end{bmatrix};$$

$$\mathbf{A}(G_3)=\begin{bmatrix}1 & 5 & 1 & 7\\ 1/5 & 1 & 1/5 & 3\\ 1 & 5 & 1 & 7\\ 1/7 & 1/3 & 1/7 & 1\end{bmatrix};\quad \mathbf{A}(G_4)=\begin{bmatrix}1 & 1/3 & 1/5 & 1/3\\ 3 & 1 & 1/2 & 1\\ 5 & 2 & 1 & 3\\ 3 & 1 & 1/3 & 1\end{bmatrix};$$

$$\mathbf{A}(G_5)=\begin{bmatrix}1 & 1/3 & 1/3 & 1/5\\ 3 & 1 & 1 & 1/3\\ 3 & 1 & 1 & 1/2\\ 5 & 3 & 2 & 1\end{bmatrix};\quad \mathbf{A}(G_6)=\begin{bmatrix}1 & 1/7 & 1/3 & 1/7\\ 7 & 1 & 3 & 1\\ 3 & 1/3 & 1 & 1/3\\ 7 & 1 & 3 & 1\end{bmatrix}. \qquad (1)$$

У кожній матриці шість елементів відповідають парним порівнянням з табл.1.1. Останні елементи знайдені з урахуванням того, що матриця парних порівнянь є діагональною і обернено симетричною.

Застосовуючи формули

$$\mu(u_i) = w_i\,,\ i=\overline{1,n}. \qquad (2)$$

і

$$\begin{cases} AW = \lambda_{\max} W, \\ w_1 + w_2 + \ldots + w_n = 1, \end{cases} \qquad (3)$$

до матриць парних порівнянь (1), отримуємо наступні нечіткі множини:

$$\tilde{G}_1=\left\{\frac{0,39}{P_1},\frac{0,39}{P_2},\frac{0,15}{P_3},\frac{0,07}{P_4}\right\};\quad \tilde{G}_2=\left\{\frac{0,59}{P_1},\frac{0,22}{P_2},\frac{0,12}{P_3},\frac{0,07}{P_4}\right\};$$

$$\tilde{G}_3=\left\{\frac{0,42}{P_1},\frac{0,11}{P_2},\frac{0,42}{P_3},\frac{0,05}{P_4}\right\};\quad \tilde{G}_4=\left\{\frac{0,08}{P_1},\frac{0,23}{P_2},\frac{0,48}{P_3},\frac{0,21}{P_4}\right\};$$

$$\tilde{G}_5=\left\{\frac{0,08}{P_1},\frac{0,23}{P_2},\frac{0,48}{P_3},\frac{0,21}{P_4}\right\};\quad \tilde{G}_6=\left\{\frac{0,06}{P_1},\frac{0,40}{P_2},\frac{0,14}{P_3},\frac{0,40}{P_4}\right\}. \qquad (4)$$

Із (4) випливає, що проект P_1 є кращим за критеріями G_1 , G_2 і G_3 , проект P_2 – за критеріями G_1 і G_6 , проект P_3 – за критеріями G_3 , G_4 і G_5 , проект P_4 – за критерієм G_6 .

Для розрахунку коефіцієнтів відносної важливості критеріїв скористаємося експертним методом парних порівнянь. Вважатимемо відомими наступні лінгвістичні парні порівняння важливості критеріїв:

· *майже істотна* перевага G_1 над G_4;

· *відсутня* перевага G_1 над G_5;

· *слабка* перевага G_1 над G_6;

· *слабка* перевага G_2 над G_1;

· *майже слабка* перевага G_2 над G_3;

· *майже сильна* перевага G_2 над G_4;

· *слабка* перевага G_2 над G_5;

· *істотна* перевага G_2 над G_6;

· *майже слабка* перевага G_3 над G_1;

· *істотна* перевага G_3 над G_4;

· *майже слабка* перевага G_3 над G_5;

· *слабка* перевага G_3 над G_6;

· *слабка* перевага G_5 над G_4;

· *майже слабка* перевага G_5 над G_6;

· *майже слабка* перевага G_6 над G_4.

Експертним висловам відповідає така матриця парних порівнянь:

$$\mathbf{A} = \begin{bmatrix} 1 & 1/3 & 1/2 & 4 & 1 & 3 \\ 3 & 1 & 2 & 6 & 3 & 5 \\ 2 & 1/2 & 1 & 5 & 2 & 3 \\ 1/4 & 1/6 & 1/5 & 1 & 1/3 & 1/2 \\ 1 & 1/3 & 1/2 & 3 & 1 & 2 \\ 1/3 & 1/5 & 1/3 & 2 & 1/2 & 1 \end{bmatrix}.$$

За формулами (2) і (3) знаходимо коефіцієнти відносної важливості критеріїв G_1 , G_2 , …, G_6 :

$$\alpha_1 = 0{,}15; \quad \alpha_2 = 0{,}34; \quad \alpha_3 = 0{,}26; \quad \alpha_4 = 0{,}05; \quad \alpha_5 = 0{,}13; \quad \alpha_6 = 0{,}07,$$

що означає найбільшу важливість при прийнятті рішення очікуваного ефекту (G_1) і ризиків (G_3).

За формулою

$$\mu_D(P_j) = \min_{i=\overline{1,n}} (\mu_{G_i}(P_j))^{\alpha_i}, \quad j = \overline{1, k},$$

де a_i – коефіцієнт відносної важливості критерія G_i, $a_1 + a_2 + \ldots + a_n = 1$.

розрахуємо такі нечіткі множини:

$$\tilde{G}_1^{\alpha_1} = \left\{ \frac{0{,}39^{0{,}15}}{P_1}, \frac{0{,}39^{0{,}15}}{P_2}, \frac{0{,}15^{0{,}15}}{P_3}, \frac{0{,}07^{0{,}15}}{P_4} \right\} = \left\{ \frac{0{,}868}{P_1}, \frac{0{,}868}{P_2}, \frac{0{,}753}{P_3}, \frac{0{,}667}{P_4} \right\};$$

$$\tilde{G}_2^{\alpha_2} = \left\{ \frac{0{,}59^{0{,}34}}{P_1}, \frac{0{,}22^{0{,}34}}{P_2}, \frac{0{,}12^{0{,}34}}{P_3}, \frac{0{,}07^{0{,}34}}{P_4} \right\} = \left\{ \frac{0{,}835}{P_1}, \frac{0{,}596}{P_2}, \frac{0{,}490}{P_3}, \frac{0{,}409}{P_4} \right\};$$

$$\tilde{G}_3^{\alpha_3} = \left\{ \frac{0{,}42^{0{,}26}}{P_1}, \frac{0{,}11^{0{,}26}}{P_2}, \frac{0{,}42^{0{,}26}}{P_3}, \frac{0{,}05^{0{,}26}}{P_4} \right\} = \left\{ \frac{0{,}797}{P_1}, \frac{0{,}552}{P_2}, \frac{0{,}797}{P_3}, \frac{0{,}456}{P_4} \right\};$$

$$\tilde{G}_4^{\alpha_4} = \left\{ \frac{0{,}08^{0{,}05}}{P_1}, \frac{0{,}23^{0{,}05}}{P_2}, \frac{0{,}48^{0{,}05}}{P_3}, \frac{0{,}21^{0{,}05}}{P_4} \right\} = \left\{ \frac{0{,}894}{P_1}, \frac{0{,}936}{P_2}, \frac{0{,}969}{P_3}, \frac{0{,}933}{P_4} \right\};$$

$$\tilde{G}_5^{\alpha_5} = \left\{ \frac{0{,}08^{0{,}13}}{P_1}, \frac{0{,}23^{0{,}13}}{P_2}, \frac{0{,}48^{0{,}13}}{P_3}, \frac{0{,}21^{0{,}13}}{P_4} \right\} = \left\{ \frac{0{,}717}{P_1}, \frac{0{,}813}{P_2}, \frac{0{,}823}{P_3}, \frac{0{,}909}{P_4} \right\};$$

$$\tilde{G}_6^{\alpha_6} = \left\{ \frac{0{,}06^{0{,}07}}{P_1}, \frac{0{,}40^{0{,}07}}{P_2}, \frac{0{,}14^{0{,}07}}{P_3}, \frac{0{,}40^{0{,}07}}{P_4} \right\} = \left\{ \frac{0{,}813}{P_1}, \frac{0{,}938}{P_2}, \frac{0{,}871}{P_3}, \frac{0{,}938}{P_4} \right\}.$$

Перетин цих нечітких множин дає такі міри приналежності нечіткого рішення D

$$\mu_D(P_1) = \min(0{,}868,\ 0{,}835,\ 0{,}797,\ 0{,}894,\ 0{,}717,\ 0{,}813) = 0{,}717;$$

$$\mu_D(P_2) = \min(0{,}868,\ 0{,}596,\ 0{,}552,\ 0{,}936,\ 0{,}813,\ 0{,}938) = 0{,}552;$$

$$\mu_D(P_3) = \min(0{,}753,\ 0{,}490,\ 0{,}797,\ 0{,}969,\ 0{,}823,\ 0{,}871) = 0{,}490;$$

$$\mu_D(P_4) = \min(0{,}667,\ 0{,}409,\ 0{,}456,\ 0{,}933,\ 0{,}909,\ 0{,}938) = 0{,}409.$$

В результаті отримуємо нечітку множину

$$\tilde{D} = \left\{ \frac{0{,}717}{P_1}, \frac{0{,}552}{P_2}, \frac{0{,}490}{P_3}, \frac{0{,}409}{P_4} \right\},$$

що свідчить про перевагу проекту P_1 над іншими. Таким чином, проект

P_1 краще інших одночасно задовольняє всі критерії з урахуванням їх важливості. Нечіткі множини, які показують наскільки повно проекти P_1 ¸ P_4 задовільняють критерії G_1 ¸ G_6, запишемо так:

$$\widetilde{P}_1 = \left\{ \frac{0,868}{G_1}, \frac{0,835}{G_2}, \frac{0,797}{G_3}, \frac{0,894}{G_4}, \frac{0,717}{G_5}, \frac{0,813}{G_6} \right\};$$

$$\widetilde{P}_2 = \left\{ \frac{0,868}{G_1}, \frac{0,596}{G_2}, \frac{0,552}{G_3}, \frac{0,936}{G_4}, \frac{0,813}{G_5}, \frac{0,938}{G_6} \right\};$$

$$\widetilde{P}_3 = \left\{ \frac{0,753}{G_1}, \frac{0,490}{G_2}, \frac{0,797}{G_3}, \frac{0,969}{G_4}, \frac{0,823}{G_5}, \frac{0,871}{G_6} \right\};$$

$$\widetilde{P}_4 = \left\{ \frac{0,667}{G_1}, \frac{0,409}{G_2}, \frac{0,456}{G_3}, \frac{0,933}{G_4}, \frac{0,909}{G_5}, \frac{0,938}{G_6} \right\}.$$

Графіки функцій приналежності цих нечітких множин зображені на рис.11.6. Із рисунка видно, що по маловажних критеріях G_4 і G_6 відстань між проектами незначна. В той же час по пріоритетних критеріях G_2 і G_3 різниця між проектами істотна.

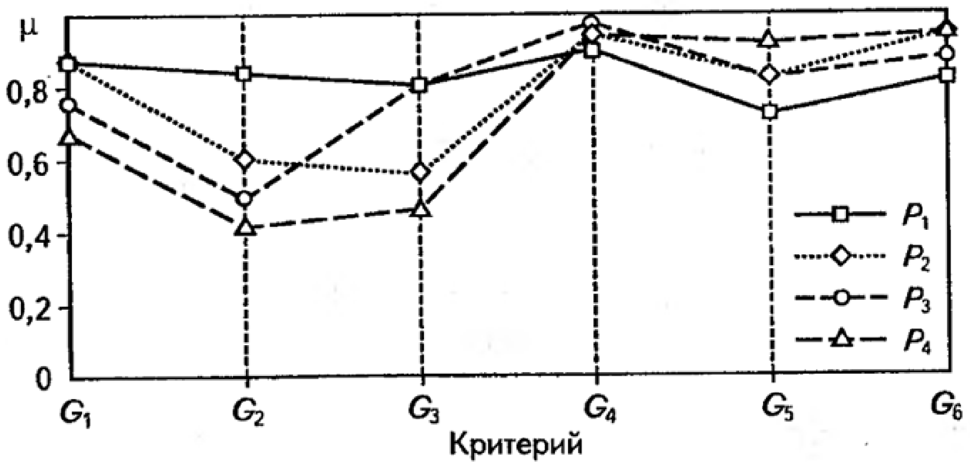


Рис.5 – Порівняння проектів P_1 ¸ P_4 з урахуванням важливості критеріїв G_1 ¸ G_6

ТЕМА 10 АНАЛІЗ ВАРІАНТІВ «ЩО БУДЕ, ЯКЩО …»

При багатокритеріальному аналізі часто виникає питання: «Що необхідно змінити в деякій альтернативі, аби вона стала найкращою?» Для відповіді на нього треба знати, наскільки чутливо прийняте рішення до експертних парних порівнянь. Розглянемо методику аналізу чутливості, ідея якої полягає у визначенні того, *яке* буде рішення, *якщо* змінити одне з парних порівнянь.

При зміні одного з парних порівнянь варіантів необхідно забезпечити несуперечність останніх. Наприклад, змінюється a_{ij} –рівень переваги варіанту P_i над варіантом P_j . Тоді в матриці парних порівнянь необхідно змінити і елемент a_{ji} , бо вони пов'язані залежністю a_{ij} =1/a_{ji}. Крім того, можливі зміни значень в рівнях переваги варіанту P_i над іншими, яким відповідають елементи a_{ir} і a_{ri} =1/ a_{ir} матриці парних порівнянь. Нижче розглядаються чотири ситуації, коли нове значення елементу вимагає корегування елементу a_{ij} матриці парних порівнянь.

1. Нехай перевага варіанту P_i над варіантом P_j сильніша, чим перевага варіанту P_i над P_r , тобто $a_{ij} < a_{rj}$ (або $a_{ji} > a_{jr}$). Тоді в безпосередньому парному порівнянні варіант P_i не повинен перевершувати варіант P_r , відповідно $a_{ir} \leq 1$. Математично запишемо це таким правилом:

$$\text{если } a_{ji} > a_{jr}, \text{ тогда } a_{ir} := \min(1,\ a_{ir}). \qquad (1)$$

2. Нехай перевага варіанту P_i над варіантом P_j сильніша, чим перевага варіанту P_r над P_j, тобто $a_{ij} > a_{rj}$. Тоді в безпосередньому парному порівнянні варіант P_r не повинен перевершувати варіант P_i , відповідно $a_{ir} \geq 1$. Математично запишемо це таким правилом:

$$\text{если } a_{ij} > a_{rj}, \text{ тогда } a_{ir} := \max(1,\ a_{ir}). \qquad (2)$$

3. Нехай варіант P_i кращий за P_j, ($a_{ij} > 1$), а варіант P_j кращий за P_r ($a_{jr} \geq$ 1). Тоді варіант P_i буде кращим за P_r , відповідно $a_{ir} > 1$. При цьому a_{ir} – рівень переваги P_i над P_r – повинен бути не менше, чим a_{ij} та a_{jr}. Запишемо

це таким правилом:

$$\text{если } a_{ij} > 1 \text{ и } a_{jr} > 1, \text{ тогда } a_{ir} := \max(a_{ij}, a_{jr}, a_{ir}). \quad (3)$$

4. Нехай варіант P_i гірший за P_j , ($a_{ij} < 1$), а варіант P_j гірший за P_r ($a_{jr} < 1$). Тоді варіант P_i повинен бути гіршим за P_r , відповідно $a_{ir} < 1$. При цьому a_{ri} – рівень переваги P_r над P_i – повинен бути не менше, чим $a_{ji} = 1/a_{ij}$ та $a_{rj} = 1/a_{jr}$. Запишемо це таким правилом:

$$\text{если } a_{ij} < 1 \text{ и } a_{jr} < 1, \text{ тогда } a_{ir} := \min(a_{ij}, a_{jr}, a_{ir}). \quad (4)$$

Нижче наводиться покрокова методика «що буде, якщо …» аналізу варіантів, яка використовує правила (1) – (4).

Крок 1. Позначити аналізований варіант P_i .

Крок 2. Виявити критерій, по якому можна поліпшити варіант P_i та позначити цей критерій через G_u .

Крок 3. Визначити варіант, з яким зручно порівнювати варіант P_i по критерію G_u. Позначити цей варіант-аналог через P_j.

Крок 4: Змінити за шкалою Сааті значення елементу a_{ij} в матриці парних порівнянь $\mathbf{A}(G_u)$.

Крок 5. Розрахувати значення елементу a_{ji} в матриці парних порівнянь $\mathbf{A}(G_u)$ по формулі $a_{ji} = 1/a_{ij}$.

Крок 6. Перерахувати значення елементів a_{ir} та a_{rj} (, $r^1 i$, $r^1 j$) по правилах (1) – (4).

Крок 7. Забезпечити обернену симетричність матриці $\mathbf{A}(G_u)$.

Крок 8. Розрахувати нові міри приналежності нечіткої множини.

Крок 9. Провести багатокритеріальний аналіз варіантів по формулах (1) –(4) і зафіксувати прийняте рішення.

Крок 10. Повторити кроки 4-9 для всіх можливих парних порівнянь варіантів P_i та P_j по критерію G_u .

Приклад 1. Дані для багатокритеріального аналізу бренд-проектів приведені в попередньому підрозділі. Необхідно встановити, яким має бути проект P_3 аби він став найкращим.

Проект P_3 має третій ранг; проекти P_1 і P_2 кращі за нього. Передбачимо, що можна поліпшити проект P_3 по критерію G_2 . Промоделюємо, як вплине на прийняття рішення зміна рівня переваги проекту P_3 над P_1 з поточного значення «істотна перевага P_1 над P_3» до оцінки «слабка перевага P_3 над P_1». Для цього поміняємо значення елементу a_{31} матриці парних порівнянь $\mathbf{A}(G_2)$ з 1/5 на 1/4, 1/3, 1/2, 1, 2 і 3, і проведемо розрахунки по описаній вище методиці. Результати розрахунків зведені в таблицю 1 і зображені графіками залежності рішення від зміни парного порівняння a_{31} (рис.1).

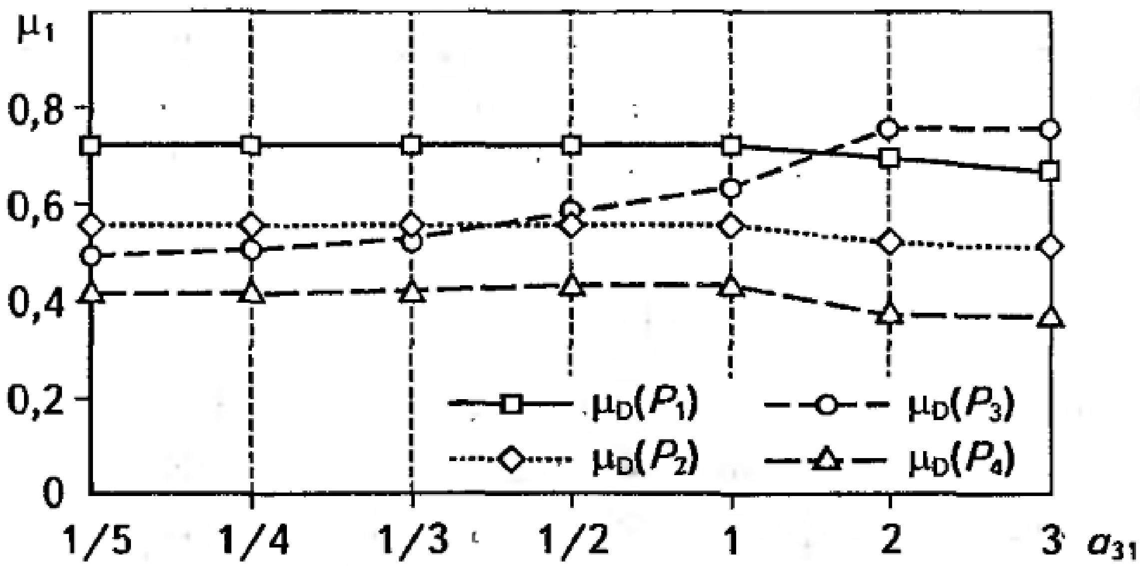


Рис.11.7 – Результати «що буде, якщо …» аналізу бренд-проектів

З рисунка видно, що проект P_3 стане другим по рангу, коли по критерію G_2 перевага P_1 над P_3 буде менше слабкої (a_{31} > 1/3). Проект P_3 стане найкращим, коли він буде хоч трохи перевершувати проект P_1 по критерію G_2 (a_{31} > 1).

Таблиця 1 – Розрахунки залежності прийняття рішення від зміни парного порівняння a_{31} по критерію G_2

a_{31}	$\mathbf{A}(G_2)$	$\widetilde{G}_2$	$\widetilde{D}$	D
1/5	$\begin{bmatrix} 1 & 3 & 5 & 7 \\ 1/3 & 1 & 2 & 3 \\ 1/5 & 1/2 & 1 & 2 \\ 1/7 & 1/3 & 1/2 & 1 \end{bmatrix}$	$\left\{\frac{0,59}{P_1}, \frac{0,22}{P_2}, \frac{0,12}{P_3}, \frac{0,07}{P_4}\right\}$	$\left\{\frac{0,717}{P_1}, \frac{0,552}{P_2}, \frac{0,490}{P_3}, \frac{0,409}{P_4}\right\}$	P_1
1/4	$\begin{bmatrix} 1 & 3 & 4 & 7 \\ 1/3 & 1 & 2 & 3 \\ 1/4 & 1/2 & 1 & 2 \\ 1/7 & 1/3 & 1/2 & 1 \end{bmatrix}$	$\left\{\frac{0,57}{P_1}, \frac{0,23}{P_2}, \frac{0,13}{P_3}, \frac{0,07}{P_4}\right\}$	$\left\{\frac{0,717}{P_1}, \frac{0,552}{P_2}, \frac{0,504}{P_3}, \frac{0,412}{P_4}\right\}$	P_1
1/3	$\begin{bmatrix} 1 & 3 & 3 & 7 \\ 1/3 & 1 & 2 & 3 \\ 1/3 & 1/2 & 1 & 2 \\ 1/7 & 1/3 & 1/2 & 1 \end{bmatrix}$	$\left\{\frac{0,55}{P_1}, \frac{0,23}{P_2}, \frac{0,15}{P_3}, \frac{0,07}{P_4}\right\}$	$\left\{\frac{0,717}{P_1}, \frac{0,552}{P_2}, \frac{0,522}{P_3}, \frac{0,415}{P_4}\right\}$	P_1
1/2	$\begin{bmatrix} 1 & 3 & 2 & 7 \\ 1/3 & 1 & 1 & 3 \\ 1/2 & 1 & 1 & 2 \\ 1/7 & 1/3 & 1/2 & 1 \end{bmatrix}$	$\left\{\frac{0,52}{P_1}, \frac{0,20}{P_2}, \frac{0,20}{P_3}, \frac{0,08}{P_4}\right\}$	$\left\{\frac{0,717}{P_1}, \frac{0,552}{P_2}, \frac{0,582}{P_3}, \frac{0,423}{P_4}\right\}$	P_1
1	$\begin{bmatrix} 1 & 3 & 1 & 7 \\ 1/3 & 1 & 1 & 3 \\ 1 & 1 & 1 & 2 \\ 1/7 & 1/3 & 1/2 & 1 \end{bmatrix}$	$\left\{\frac{0,45}{P_1}, \frac{0,21}{P_2}, \frac{0,26}{P_3}, \frac{0,08}{P_4}\right\}$	$\left\{\frac{0,717}{P_1}, \frac{0,552}{P_2}, \frac{0,631}{P_3}, \frac{0,428}{P_4}\right\}$	P_1
2	$\begin{bmatrix} 1 & 3 & 1/2 & 7 \\ 1/3 & 1 & 1/3 & 3 \\ 2 & 3 & 1 & 7 \\ 1/7 & 1/3 & 1/7 & 1 \end{bmatrix}$	$\left\{\frac{0,33}{P_1}, \frac{0,14}{P_2}, \frac{0,48}{P_3}, \frac{0,05}{P_4}\right\}$	$\left\{\frac{0,689}{P_1}, \frac{0,512}{P_2}, \frac{0,753}{P_3}, \frac{0,368}{P_4}\right\}$	P_3
3	$\begin{bmatrix} 1 & 3 & 1/3 & 7 \\ 1/3 & 1 & 1/3 & 3 \\ 3 & 3 & 1 & 7 \\ 1/7 & 1/3 & 1/7 & 1 \end{bmatrix}$	$\left\{\frac{0,30}{P_1}, \frac{0,13}{P_2}, \frac{0,52}{P_3}, \frac{0,05}{P_4}\right\}$	$\left\{\frac{0,660}{P_1}, \frac{0,505}{P_2}, \frac{0,753}{P_3}, \frac{0,364}{P_4}\right\}$	P_3

ТЕМА 11. СИНТЕЗ НЕЧІТКИХ ДИНАМІЧНИХ МОДЕЛЕЙ У MATLAB

Бібліотека нечітких блоків fuzblock для взаємодії з пакетом Simulink

Взаємодія Fuzzy Logic з Simulink відбувається через бібліотеку fuzblock (рис.12.1), що містить такі блоки...

Проектування систем керування динамічними об’єктами з нечіткими параметрами

На сучасному етапі дослідження в галузі проектування систем керування складними динамічними системами з нечіткими параметрами спрямовані на створення модульних систем нечіткого регулювання (fuzzy control system, fuzzy controller), заснованих на положеннях нечіткої логіки (fuzzy-logic), що забезпечують відповідні характеристики поведінки об’єкта керування...

Бібліотека нечітких блоків fuzblock для взаємодії з пакетом Simulink

Взаємодія Fuzzy Logic з Simulink відбувається через бібліотеку fuzblock (рис.1), що містить такі блоки:

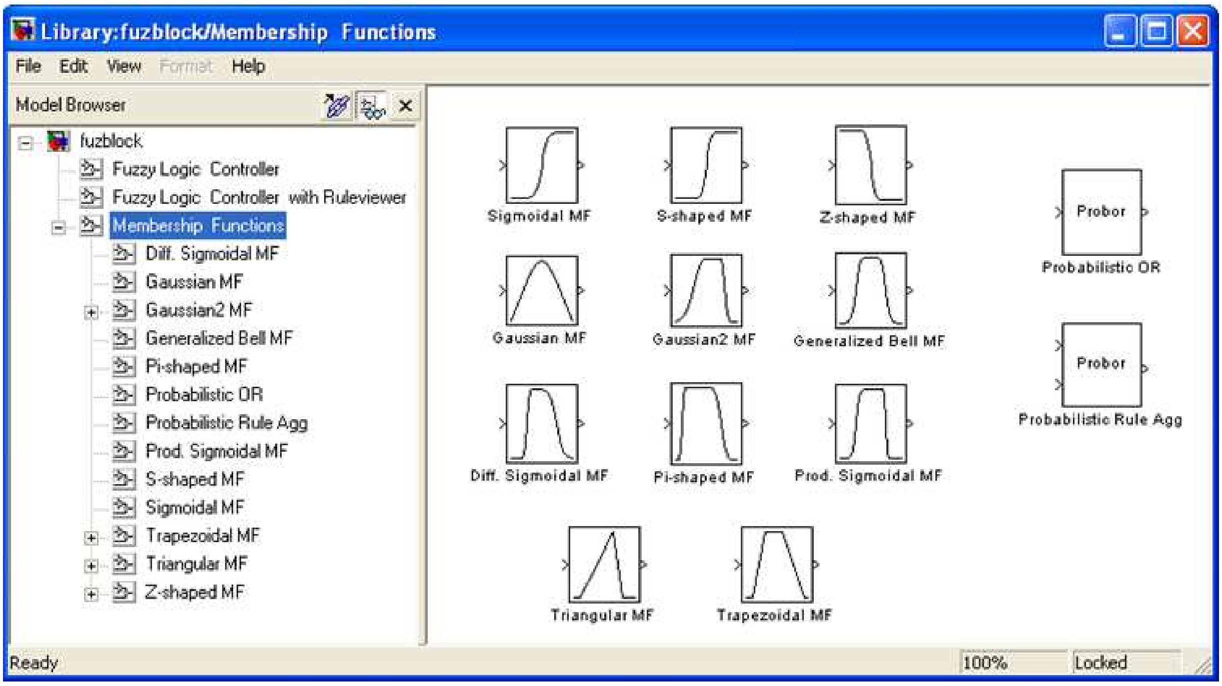


Рис.1 – Бібліотека fuzblock

• Fuzzy Logic Controller – нечіткий контролер;

• Fuzzy Logic Controller with Ruleviewer – нечіткий контролер з виведенням вікна **Rule Viewer** під час моделювання в пакеті Simulink;

• Membership Functions – бібліотека сімулінк-блоків для наступних функцій приналежностей та реалізацій логічних операцій:

• Diff. Sigmoidal MF – різниця два сигмоїдних функцій приналежності;

• Gaussian MF – гаусова функція приналежності;

• Gaussian 2MF – двостороння гаусова функція приналежності;

• Generalized Bell MF – узагальнена колоколоподібна функція приналежності;

• Pi-shaped MF – пі-подібна функція приналежності;

• Probabilistic OR – імовірнісна реалізація логічної операції АБО;

• Probabilistic Rule Agg – імовірнісна реалізація агрегації;

• Prod. Sigmoidal MF – добуток двох сигмоїдних функцій приналежності;

• S-shaped MF – S-подібна функція приналежності;

• Sigmoidal MF – сигмоїдна функція приналежності;

• Trapezoidal MF – трапецієвидна функція приналежності;

• Triangular MF – трикутна функція приналежності;

• Z-shaped MF – Z-подібна функція принадлежності.

Для включення системи нечіткого виведення в сімулінк-модуль необхідно відкрити fuzblock командою fuzblock або через опцію Fuzzy Logic Toolbox в Simulink Library Browser. Потім вибрати блок Fuzzy Logic Controller або Fuzzy Logic Controller with Ruleviewer, зробити подвійне клацання по цьому блоку і в діалоговому вікні, що з’явилося, ввести ім’я файлу або найменування змінної з цієї області, які відповідають системі нечіткого виведення. Якщо нечітка система має декілька входів, тоді в сімулинк-моделі необхідно з’єднати їх разом до введення в нечіткий контролер. Аналогічно, якщо нечітка система має декілька виходів, тоді вихідні сигнали блоку будуть представлені однією мультиплексною лінією.

Для більшості нечітких систем fuzblock автоматично генерує ієрархічну модель, що складається з сімулинк-модулей (наприклад, див.рис.2). Автоматичний синтез моделі відбувається за допомогою FIS Wizard. Синтезовані моделі складаються лише зі вбудованих сімулінк-модулей, тому нечітке виведення виконується дуже швидко, навіть якщо модель виходить громіздкою. FIS Wizard генерує сімулінк-модулі, якщо нечітка система містить лише вбудовані функції приналежності. Крім того, повинні використовуватися такі реалізації логічних операцій: АБО – max; ТА – min і prod; імплікація – min і агрегація – max. Якщо створити нечітку систему з сімулінк-блоків не вдається, тоді використовується sffis – спеціально оптимізована під Simulink функція нечіткого виведення.

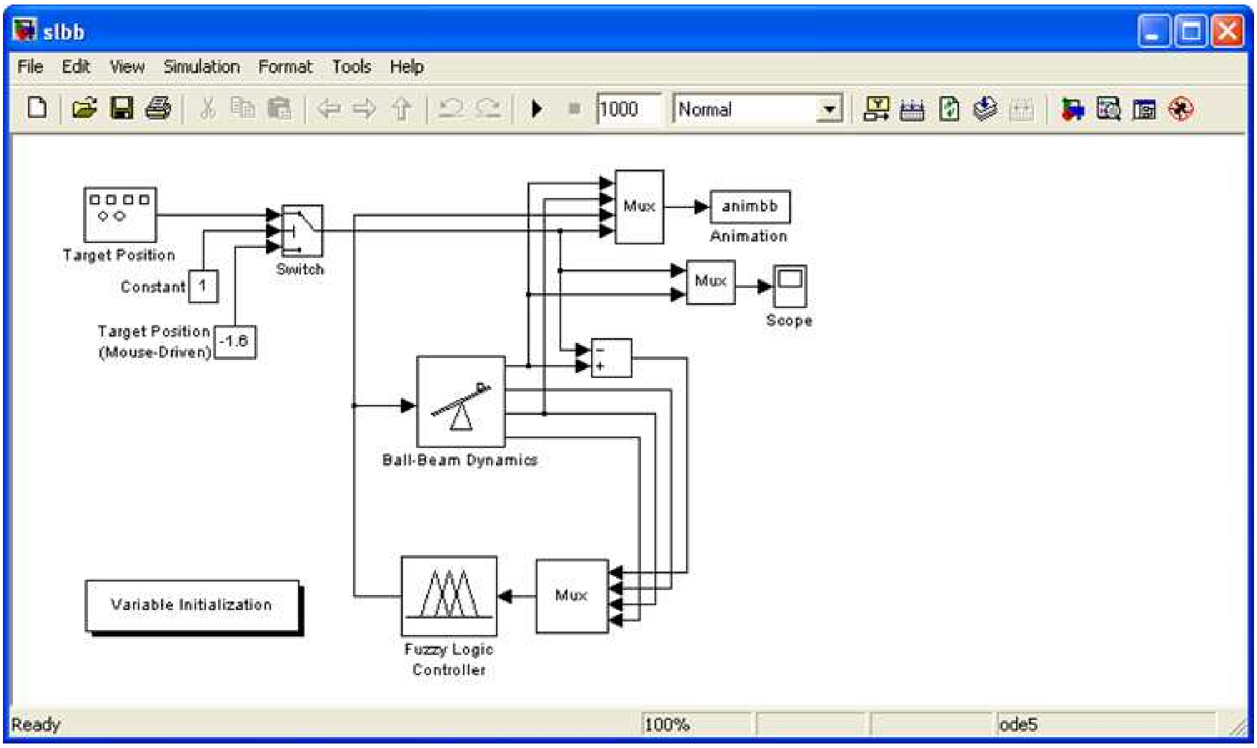


Рис.2 – Сімулінк модель системи «шарик на коромислі»

Проектування систем керування динамічними об'єктами з нечіткими параметрами

На сучасному етапі дослідження в галузі проектування систем керування складними динамічними системами з нечіткими параметрами спрямовані на створення модульних систем нечіткого регулювання (fuzzy control system, fuzzy controller), заснованих на положеннях нечіткої логіки (fuzzy-logic), що забезпечують відповідні характеристики поведінки об'єкта керування. Проектування систем автоматизованого керування нелінійними динамічними об'єктами та об'єктами з нечіткими параметрами стало можливим із розвитком сучасних інформаційних технологій та математичних методів, що доповнюються наочними геометричними інтерпретаціями.

Загальна модель нечіткого керування (рис.1), будується з урахуванням необхідності реалізації усіх етапів нечіткого висновку, а сам процес висновку реалізується на основі алгоритму нечіткого висновку.

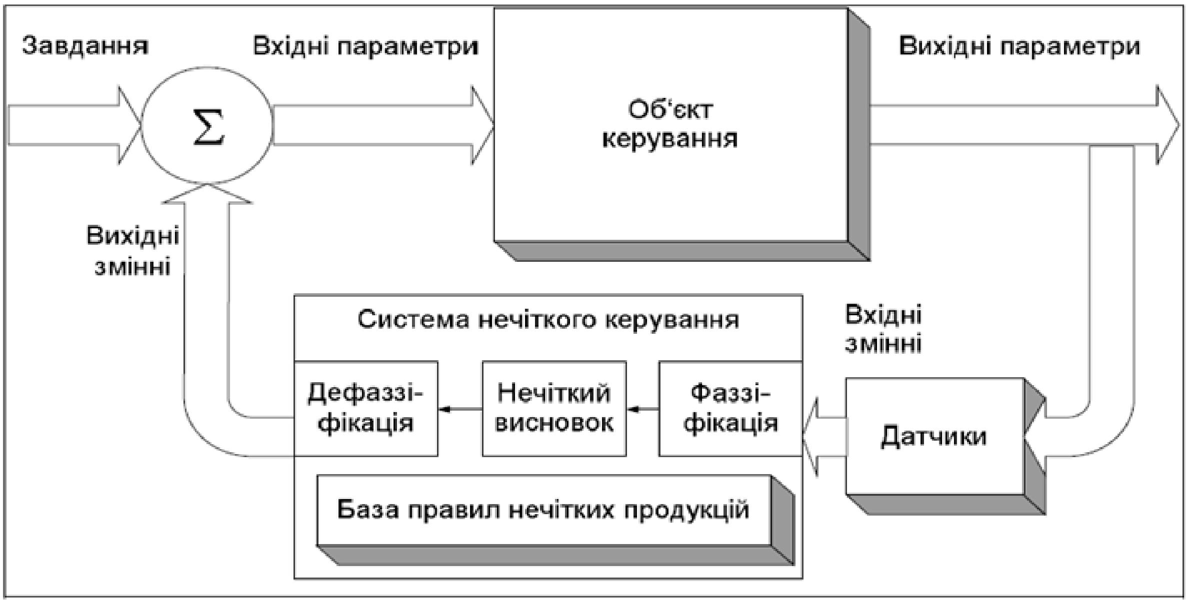


Рис.1 – Загальна схема нечіткого керування

Система нечіткого висновку призначена для перетворення значень вхідних змінних процесу керування на вихідні змінні на основі використання нечітких правил продукцій. Для цього система нечіткого висновку повинна містити базу правил нечітких продукцій та реалізовувати нечіткий висновок на основі посилок або умов, що представлені у формі нечітких лінгвістичних висловлювань. Процес переходу від нечіткого висновку до чіткого рішення наведений в алгоритмічній формі (рис.2).

Розглянемо реалізацію цього алгоритму на прикладі розробки системи керування динамічним об'єктом з нечіткими параметрами. В якості об'єкта керування виберемо систему кондиціонування повітря в приміщенні (рис.3). Оскільки температура навколишньої середи навкруг приміщення змінюється, це дестабілізує температуру повітря в приміщенні та призводить до необхідності регулювання режиму роботи кондиціонера.

Рис.2 – Алгоритмічна схема прийняття рішення

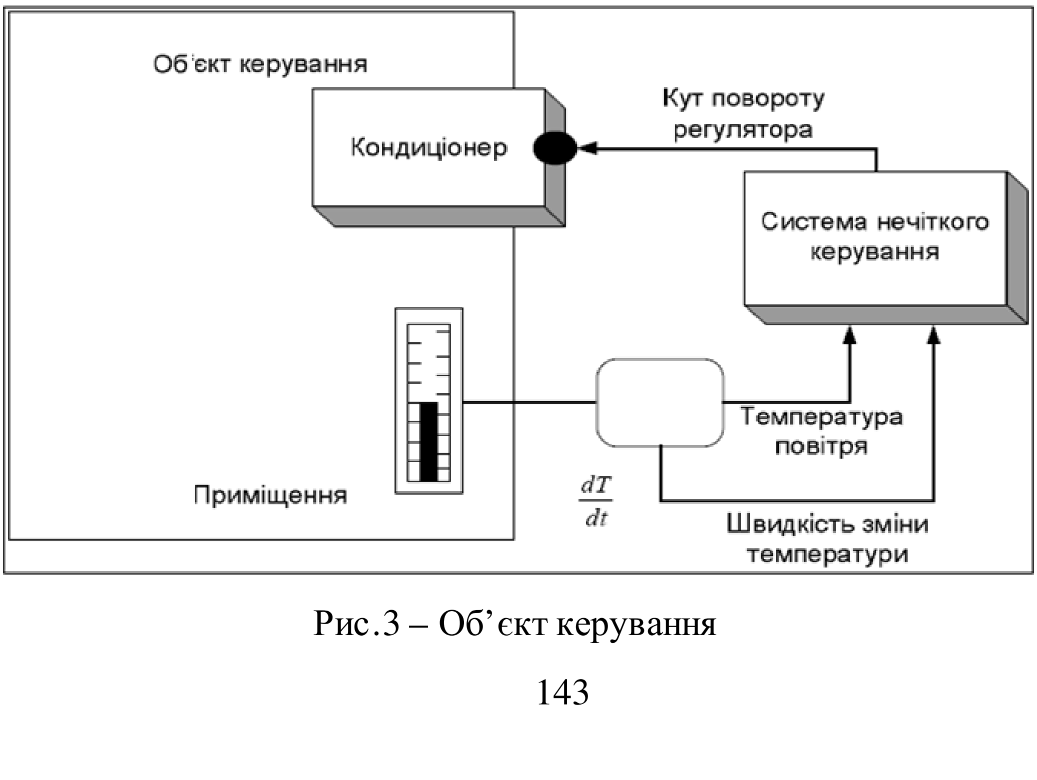


Рис.3 – Об’єкт керування

На систему нечіткого керування приходять із датчиків дві вхідні змінні: температура в приміщенні (T, °C) та швидкість її зміни ($S=dT/dt$, °C/хв).

Вихідною змінною нечіткого регулятора є поворот регулятора кондиціонера на відповідний кут відносно деякої точки, в якій кондиціонер вважається вимкненим: ліворуч для подачі холодного повітря та праворуч для подачі теплого повітря. Реалізацію цієї задачі виконаємо у середовищі MATLAB у модулі Simulink із підключенням інструментарію Fuzzy logic.

Спочатку, необхідно ввести замість чітких вхідних змінних їх лінгвістичні аналоги. Тобто, для вхідної змінної T (температура) задається набір з 5 термів (окремих нечітких значень): Cold – «холодна» температура, Normal – «нормальна» температура, Warm – «тепла» температура та Hot – «жарка» температура. Для вхідної змінної S (швидкість зміни температури) – набір з 3 термів: NS – «негативна» швидкість, Z – «нульова» швидкість, PS – «позитивна» швидкість. Для вихідної змінної *Angle* (кут повороту регулятора) – набір з 7 термів: NB – «великий негативний» кут, NM – «середній негативний» кут, NS – «малий негативний» кут, Z – «нульовий» кут, PS – «малий позитивний» кут, PM – «середній позитивний» та PB – «великий позитивний» кут. Після завдання лінгвістичних змінних переходимо до формування бази нечітких правил. База правил нечітких продукцій являє собою кінцеву множину правил нечітких продукцій, узгоджених відносно лінгвістичних змінних. Для нашого прикладу формуємо базу з 15 правил, що формують поведінку нечіткого регулятора (рис.4).

Згідно з алгоритмом нечіткого висновку, наступним етапом є процес фаззіфікації, де встановлюється відповідність між кожним конкретним значенням вхідної змінної системи нечіткого керування та відповідним до неї термом вхідної лінгвістичної змінної. Після завершення цього процесу для всіх вхідних змінних задані конкретні значення функцій належності по всім лінгвістичним термам.

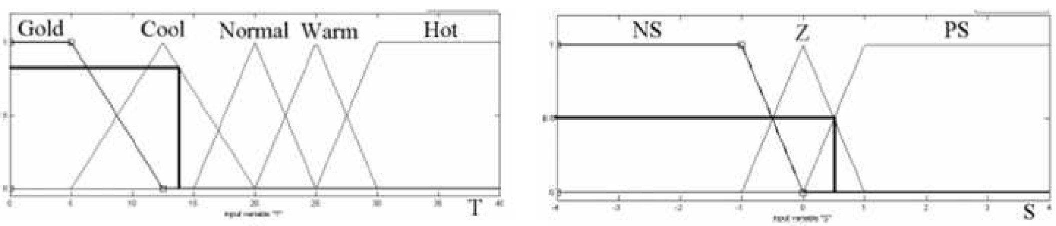
Функції належності для вхідних та вихідної змінної конкретного регулятора побудуємо на основі базових функцій пакету MATLAB. Для вхідних змінних *T* (температура) та *S* (швидкість зміни температури), а також для вихідної змінної *Angle* (кут повороту регулятора) використовуються набори трапецеїдальних та трикутних функцій належності, які мають наочну геометричну інтерпретацію.

Для прикладу візьмемо ситуацію коли температура в приміщенні дорівнює 14°C, а швидкість її зміни позитивна та дорівнює 0,5°C/хв. При даних умовах нечіткі правила починають працювати, тобто починається процес фаззіфікації.

1. If (T is Hot) and (S is PS) then (Angle is NB) (1)
2. If (T is Hot) and (S is NS) then (Angle is NS) (1)
3. If (T is Warm) and (S is PS) then (Angle is NM) (1)
4. If (T is Warm) and (S is NS) then (Angle is Z) (1)
5. If (T is Cold) and (S is NS) then (Angle is PB) (1)
6. If (T is Cold) and (S is PS) then (Angle is PS) (1)
7. If (T is Cool) and (S is NS) then (Angle is PM) (1)
8. If (T is Cool) and (S is PS) then (Angle is Z) (1)
9. If (T is Hot) and (S is Z) then (Angle is NM) (1)
10. If (T is Warm) and (S is Z) then (Angle is NS) (1)
11. If (T is Cold) and (S is Z) then (Angle is PM) (1)
12. If (T is Cool) and (S is Z) then (Angle is PS) (1)
13. If (T is Normal) and (S is PS) then (Angle is NS) (1)
14. If (T is Normal) and (S is NS) then (Angle is PS) (1)
15. If (T is Normal) and (S is Z) then (Angle is Z) (1)

Рис.4 – База правил нечітких продукцій

Процес фаззіфікації для першої вхідної змінної видає міру істинності 0,8 для терма Cool, рис.5а, а для другої вхідної змінної – 0,5 для терма Z та 0,5 для терма PS, рис.5б. Відповідні підумови використовуються в правилах нечітких продукцій з номерами 8 та 12 (рис.4, 5). Ці правила вважають активними та використовують надалі.



а) б)

Рис.5 – Процесс фаззіфікації вхідних змінних

Наступним етапом є процес агрегування підумов, тобто визначення міри істинності умов по кожному з активних нечітких правил. Агрегування підумов правила 8 дає у результаті число 0,5 (min{0,8, 0,5}), аналогічним є й результат агрегування підумов правила 12. Далі проводимо процес активізації підвисновків у нечітких правилах продукцій.

Так як коефіцієнти ваги дорівнюють 1, тоді активізація правил 8 та 12 дають дві нечіткі множини (рис.6 а, б). Акумулювання висновків, з використанням операції max-диз'юнкції, для правил 8 та 12 призводить до формування результуючої нечіткої множини (рис. 6в).

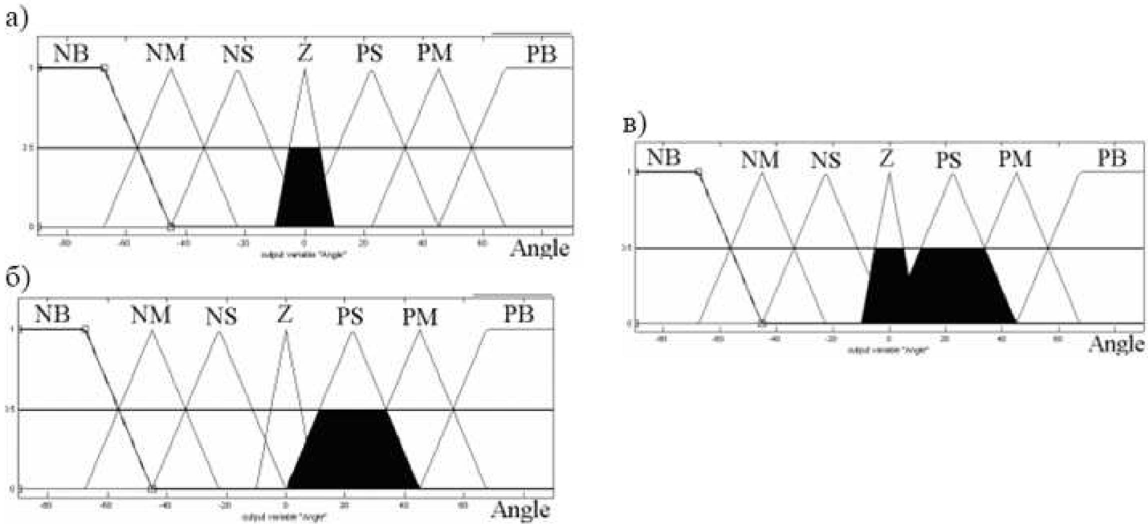


Рис.6 – Процеси активізації (а, б) та акумулювання (в) вихідної змінної

За допомогою процесу дефаззіфікації, наприклад, методом центру тяжіння, виявляється найкраща, на даний час, стратегія поведінки системи. Для конкретного прикладу маємо результат *Angle*=16,2°, тобто подачу теплого повітря в приміщення.

Усі варіанти поведінки нечіткого регулятора, при усіх можливих комбінаціях вхідних змінних, формують поверхню поведінки регулятора (рис.7), за допомогою якої є можливість досліджувати взаємозв'язки між параметрами. Крім того, є можливість дослідити поведінку нечіткого регулятора і за допомогою векторних полів (рис.8).

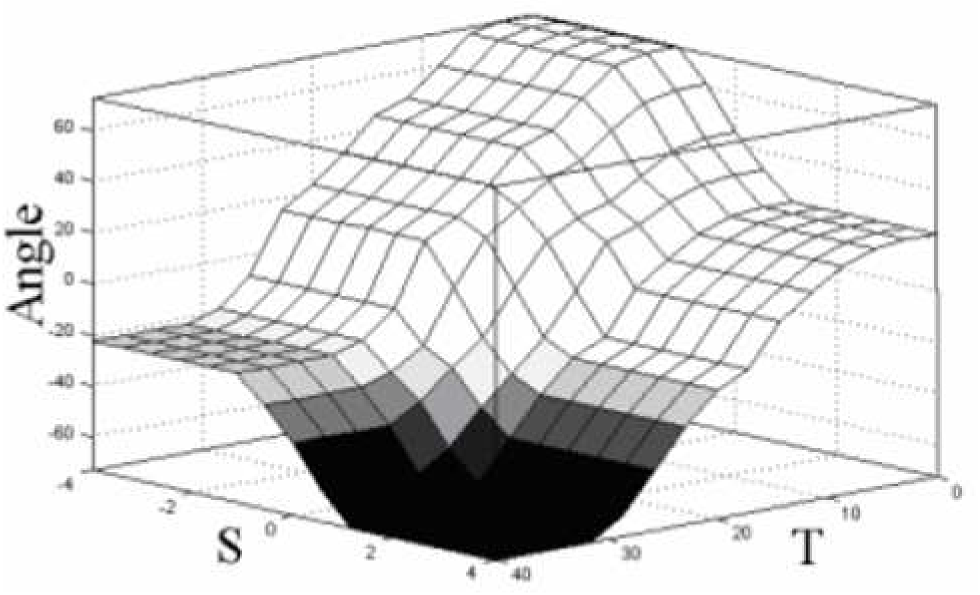


Рис.7 – Поверхня поведінки нечіткого регулятора

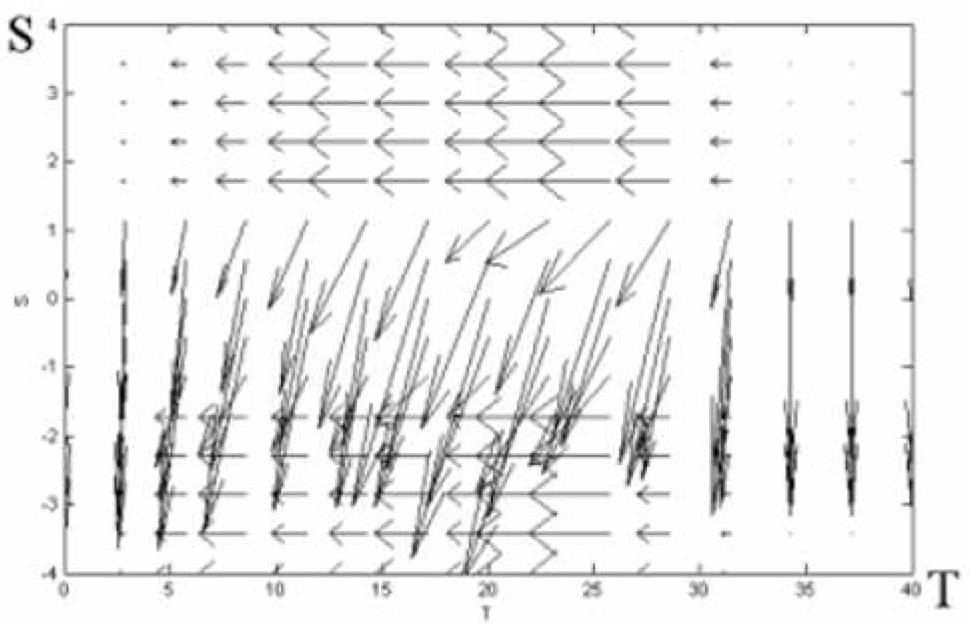


Рис.8 – Поведінка нечіткого регулятора на векторному полі

ТЕМА 12 РОЗРОБКА НЕЙРО-НЕЧІТКИХ МОДЕЛЕЙ ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ У СЕРЕДОВИЩІ MATLAB

Прогнозування часового ряду Маккея-Глесса

Розглянемо приклад вживання технології ANFIS в Matlab для прогнозування часового ряду Маккея-Глесса (Mackey-Glass). Диференціальне рівняння цього ряду має вигляд...

Аналіз і прогнозування валютних цін на фінансовому ринку

Розглянемо процес розробки нечіткої моделі гібридної мережі для вирішення завдання прогнозування валютних цін на фінансовому ринку. Суть даного завдання полягає в тому, аби, знаючи динаміку зміни курсової вартості продажу деякої валюти за фіксований інтервал часу, передбачити значення її курсової вартості на певний момент часу в майбутньому...

Прогнозування часового ряду Маккея-Глесса

Розглянемо приклад вживання технології ANFIS в Matlab для прогнозування часового ряду Маккея-Глесса (Mackey-Glass). Диференціальне рівняння цього ряду має вигляд:

$$\frac{dx(t)}{dt}=\frac{0,2x(t-\tau)}{1+x^{10}(t-\tau)}-0,1x(t).$$

При значеннях $x(0)$=1,2 і t=17 ряд виходить неперіодичним і неконвергентним, динаміка якого дуже чутлива до початкових умов. На рис.1 представлений фрагмент часового ряду Маккея-Глесса. Часовий ряд на рис.1 отриманий в припущенні, що $x(t)$=0 при t<0.

```
load mgdata.dat
a = mgdata;
time = a(:, 1);
x_t = a(:, 2);
plot(time, x_t);
```

```
xlabel('Time (sec)'); ylabel('x(t)');
title('Mackey-Glass Chaotic Time Series');
```

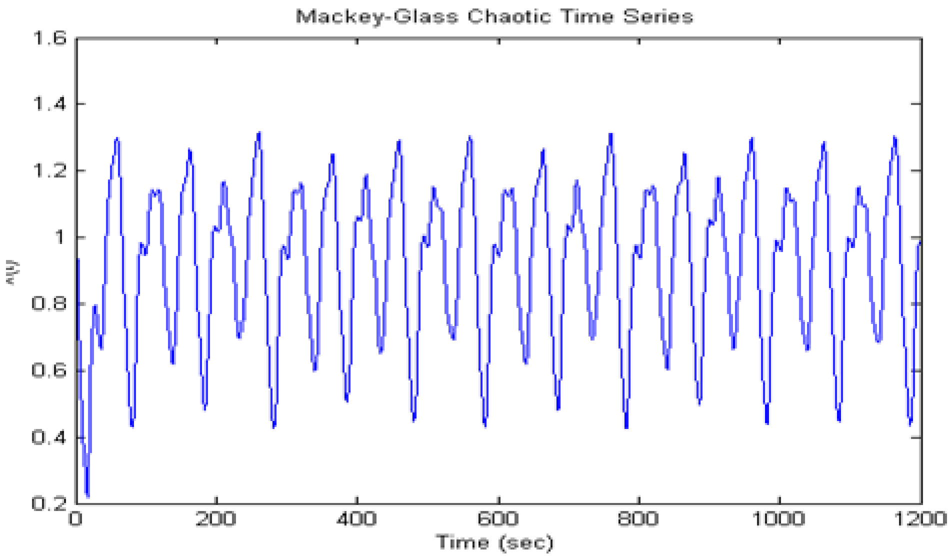


Рис.1 – Часовий ряд Маккея-Глесса

ANFIS-технологія буде використана для побудови моделі прогнозування часового ряду , що прогнозує значення ряду через шість інтервалів часу $x(t+6)$ на основі попередніх чотирьох спостережень часового ряду: $x(t-18)$, $x(t-12)$, $x(t-6)$ і $x(t)$. Отже, необхідно використовувати такий формат навчальної вибірки: [$x(t-18)$, $x(t-12)$, $x(t-6)$, $x(t)$ і $x(t+6)$]. В інтервалі від t=118 до t=1117 зібрано 1000 пар даних в указаному форматі. Перші 500 пар використовуються як навчальна вибірка, а інші – як тестова. На рис.2 показаний графік часового ряду на проміжку [118; 1117], тобто на інтервалі, з якого сформовані навчальна та тестова вибірки.

```
trn_data = zeros(500, 5);
chk_data = zeros(500, 5);

% prepare training data
trn_data(:, 1) = x_t(101:600);
trn_data(:, 2) = x_t(107:606);
trn_data(:, 3) = x_t(113:612);
trn_data(:, 4) = x_t(119:618);
```

```
trn_data(:, 5) = x_t(125:624);

% prepare checking data
chk_data(:, 1) = x_t(601:1100);
chk_data(:, 2) = x_t(607:1106);
chk_data(:, 3) = x_t(613:1112);
chk_data(:, 4) = x_t(619:1118);
chk_data(:, 5) = x_t(625:1124);

index = 119:1118; % ts starts with t = 0
plot(time(index), x_t(index));
xlabel('Time (sec)'); ylabel('x(t)');
title('Mackey-Glass Chaotic Time Series');
```

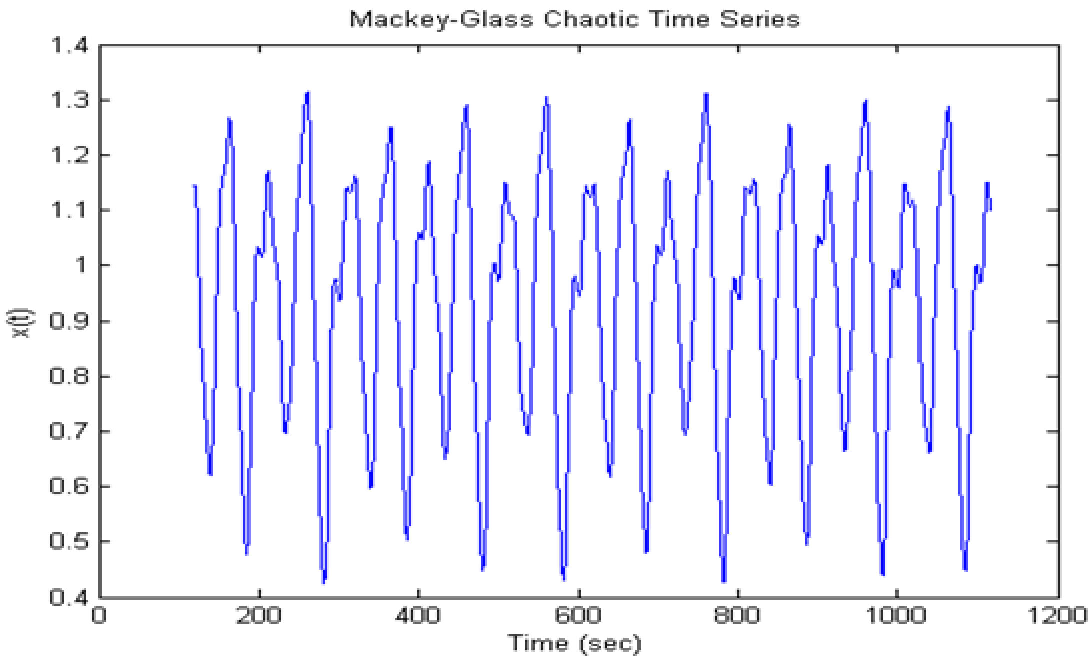


Рис.2 – Часовий ряд Маккея-Глесса на інтервалі [118; 1117]

Далі з навчальної вибірки trn_dat генерується вихідна нечітка модель по команді fismat=genfis1(trn_data). Для

лінгвістичної оцінки вхідних змінних використовується по два терми з колоколоподібними функціями належності, графіки яких приведені на рис.3а.

```
fismat = genfis1(trn_data);
 % The initial MFs for training are shown in the plots.
for input_index=1:4,
 subplot(2,2,input_index)
 [x,y]=plotmf(fismat,'input',input_index);
 plot(x,y)
 axis([-inf inf 0 1.2]);
 xlabel(['Input ' int2str(input_index)]);
end
```

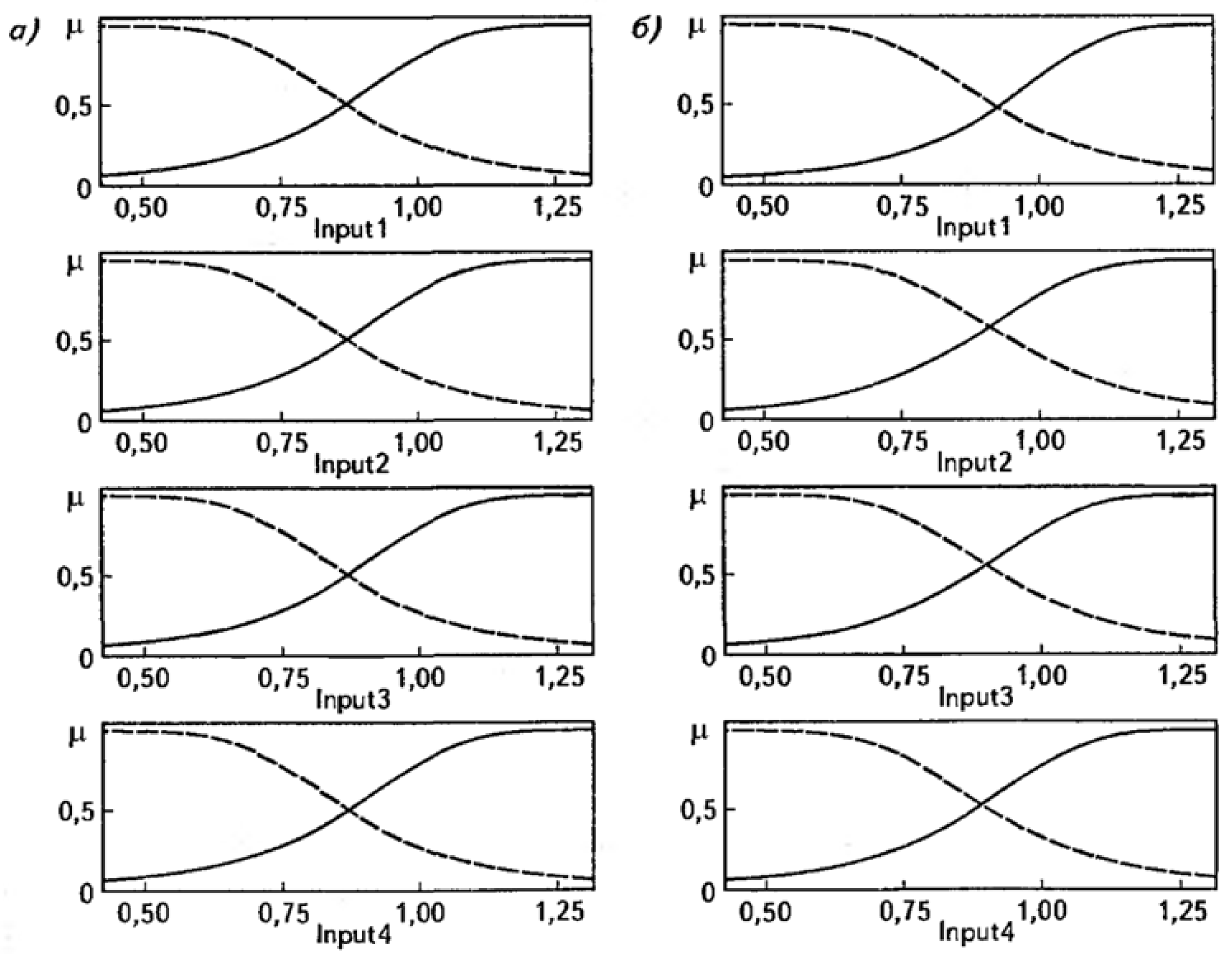


Рис.3 – Функції належності нечіткої моделі

прогнозування часового ряду

а – до навчання; *б* – після навчання

Згенерована нечітка база знань містить 2^4 = 16 правил. Кількість параметрів нечіткої моделі, що настроюються, дорівнює 104: 24 нелінійних і 80 лінійних. Таке число параметрів, що настроюються, добре збалансоване з об'ємом навчальної вибірки, яка становить 500 пар «входи – вихід». Навчання нечіткої моделі здійснюється за командою:

```
[trn_fismat, trn_error] = anfis(trn_data, [], [], chk_data).
```

На десять ітерацій навчання витрачається 4 хв. на комп'ютері SUN SPARC II Workstation.

На рис.3б приведені графіки оптимальних функцій

належності, отримані після 10 ітерацій алгоритму навчання. Вони не сильно змінилися в порівнянні з вихідними функціями належності. Вочевидь, що основне навчання сталося шляхом зміни лінійних параметрів нечіткої моделі. Нелінійні параметри зазвичай сильно змінюються при тривалому налаштуванні, яке може бути виконана пізніше.

```
% load training results
load mganfis
% plot final MF's on x, y, z, u
for input_index=1:4,
 subplot(2,2,input_index)
 [x,y]=plotmf(trn_fismat,'input',input_index);
 plot(x,y)
 axis([-inf inf 0 1.2]);
 xlabel(['Input ' int2str(input_index)]);
end
```

На рис.4 показана динаміка навчання нечіткої моделі – графіки зміни помилки RMSE на навчальній (1) і тестовій (2) вибірках.

```
% error curves plot
close all;
epoch_n = 10;
```

```
plot([trn_error chk_error]);
hold on; plot([trn_error chk_error], 'o'); hold off;
xlabel('Epochs');
ylabel('RMSE (Root Mean Squared Error)');
title('Error Curves');
```

Звернемо увагу, що помилка на тестовій вибірці менша, ніж на навчальній. Такий ефект нетиповий не лише для ANFIS-навчання, але і для нелінійної апроксимації взагалі. Можливо, процес навчання ще далекий від завершення.

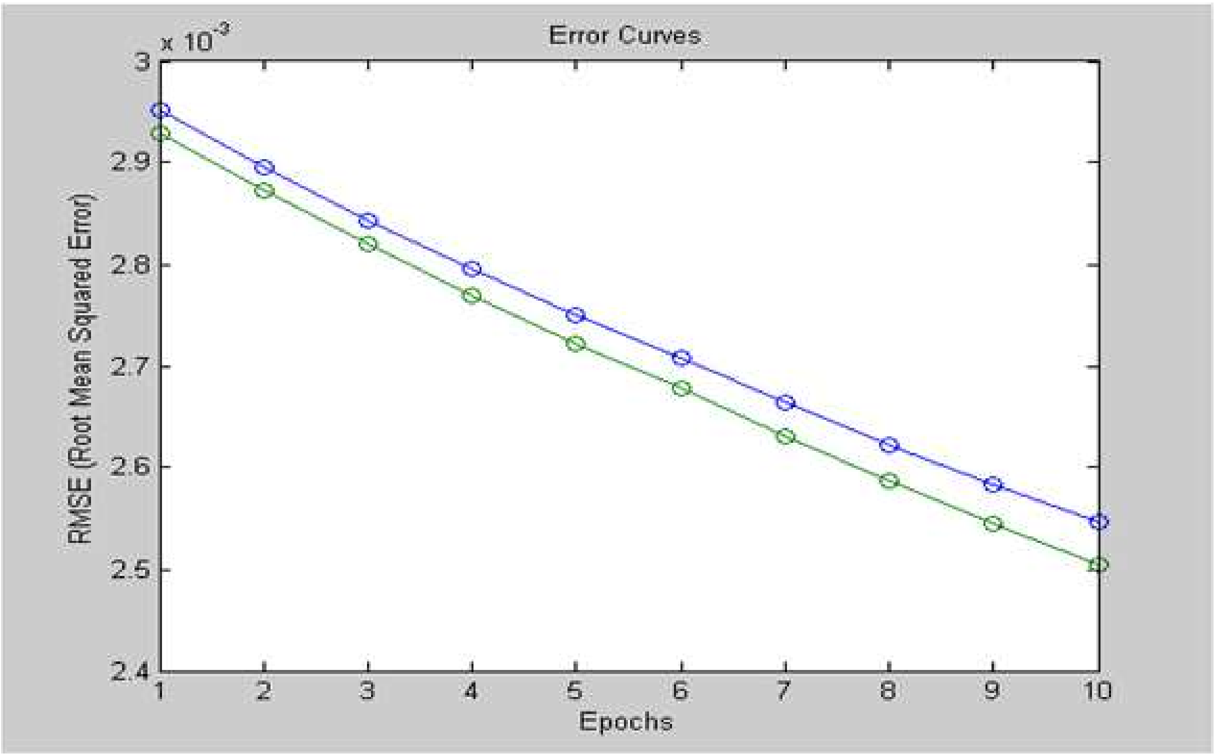


Рис.4 – Динаміка навчання нечіткої моделі прогнозування часового ряду

На рис.5 показані вихідний часовий ряд Маккея-Глесса та ряд, що спрогнозований навченою нечіткой моделлю. Різниця між графіками на рис.5 настільки незначна, що візуально замітити її неможливо (ми бачимо лише один графік). Для спостерігання похибки прогнозу необхідно змінити масштаб.

```
input = [trn_data(:, 1:4); chk_data(:, 1:4)];
anfis_output = evalfis(input, trn_fismat);
index = 125:1124;
plot(time(index), [x_t(index) anfis_output]);
xlabel('Time (sec)');
```

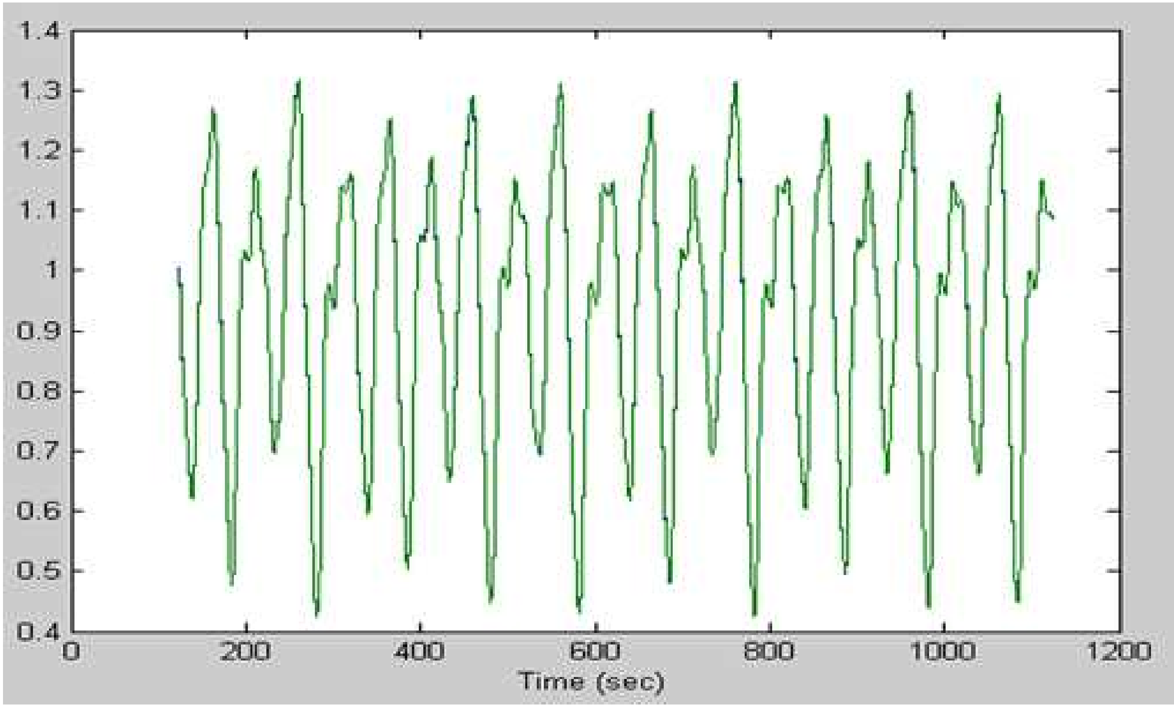


Рис.5 – Вихідний і спрогнозований часові ряди

На рис.6 показаний графік похибки прогнозу навченої нечіткої моделі (рис.6).

```
diff = x_t(index)-anfis_output;
plot(time(index), diff);
xlabel('Time (sec)');
title('ANFIS Prediction Errors');
```

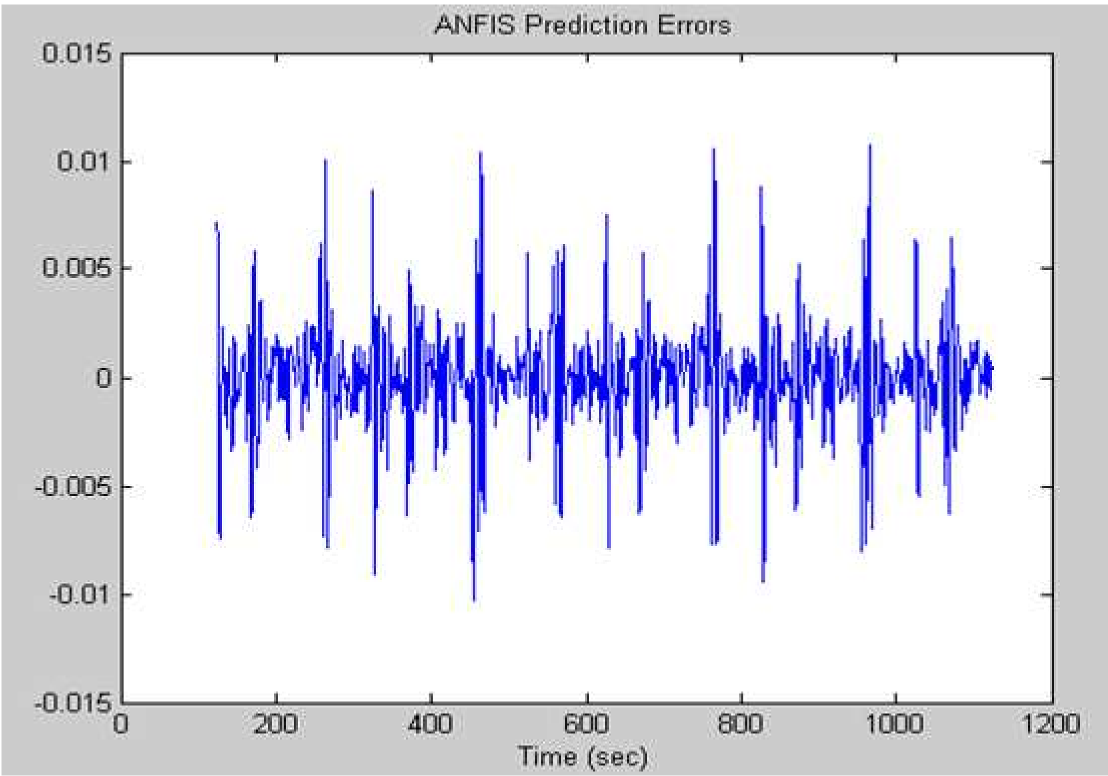


Рис.6 – Похибка прогнозу часового ряду нечіткою моделлю після навчання

Звернемо увагу, що масштаб цього слайду майже в 100 разів більше попереднього. Нагадаємо, що навчання нечіткої моделі булоперерване після 10 ітерацій. Очікується, що помилку прогноза можна зменшити подальшим навчанням нечіткої моделі.

Аналіз і прогнозування валютних цін на фінансовому ринку

Розглянемо процес розробки нечіткої моделі гібридної мережі для вирішення завдання прогнозування валютних цін на фінансовому ринку.

Суть даного завдання полягає в тому, аби, знаючи динаміку зміни курсової вартості продажу деякої валюти за фіксований інтервал часу, передбачити значення її курсової вартості на певний момент часу в майбутньому. При цьому характерною особливістю динаміки зміни курсу (тренду) є наявність двох основних тенденцій в коливаннях відповідних цін.

З одного боку, спостерігається загальне довгострокове підвищення курсової вартості, пов’язане з величиною інфляції. З іншого боку, спостерігається короткострокове коливання цін, пов’язане з цілим рядом випадкових чинників, адекватне представлення яких в тій або іншій формальній моделі навряд чи можливо.

Традиційно для вирішення даного завдання застосовуються різні моделі технічного аналізу, засновані на використанні різних індикаторів. В той же час наявність неявних тенденцій в динаміці зміни курсової вартості валют дозволяє застосувати модель адаптивних нейро-нечетких мереж.

В якості вихідних даних можна скористатися інформацією про динаміку курсу Центробанку Російської Федерації по валюті "Українська гривня" (UAH) за деякий часовий інтервал, яка доступна в Інтернеті за адресою: www.val.ru. Для конкретності візьмемо значення курсової вартості UAH за 10 одиниць в період з 1 жовтня 2008г. по 10 грудня 2008г (див.табл.1).

Таблиця 1 – Динаміка курса UAH в період з 01.10.08 по 10.12.08

Дата	Курс за 10 UAH до злотого	Дата	Курс за 10 UAH до злотого	Дата	Курс за 10 UAH до злотого
01.10.08	50.0529	01.11.08	45.2767	02.12.08	37.1555
02.10.08	50.4827	02.11.08	45.7421	03.12.08	38.1178
03.10.08	50.2849	06.11.08	46.2053	04.12.08	37.3528
04.10.08	49.7824	07.11.08	46.3590	05.12.08	37.2768
07.10.08	50.1502	08.11.08	46.5588	06.12.08	37.4555
08.10.08	48.6149	11.11.08	46.6907	09.12.08	37.0672
09.10.08	46.9711	12.11.08	47.3418	10.12.08	37.2379
10.10.08	45.9374	13.11.08	47.4856		
11.10.08	48.6234	14.11.08	47.8727		
14.10.08	50.7012	15.11.08	47.3806		
15.10.08	49.6897	18.11.08	47.5518		

16.10.08	50.8952	19.11.08	46.4917		
17.10.08	51.2519	20.11.08	45.8501		
18.10.08	49.9059	21.11.08	44.6502		
21.10.08	49.6307	22.11.08	44.1064		
22.10.08	48.0977	25.11.08	41.5959		
23.10.08	48.5072	26.11.08	41.5020		
24.10.08	46.9205	27.11.08	39.6468		
25.10.08	45.1746	28.11.08	40.0315		
28.10.08	45.9676	29.11.08	37.4319		
29.10.08	45.1269				
30.10.08	43.3566				
31.10.08	37.6230				

Передбачимо, що нечітка модель гібридної мережі міститиме 4 вхідних змінних. При цьому перша вхідна змінна відповідатиме курсу UAH на поточний банківський день, друга – курсу UAH на попередній банківський день, тобто на день (i-1), де через i позначений поточний банківський день. Тоді третя вхідна змінна відповідатиме курсу UAH на (i-2) банківський день, а четверта – курсу UAH на (i-3) банківський день.

Відповідні навчальні дані можуть бути зведені в окрему табл.2. Об’єм отриманої таким чином навчальної вибірки дорівнює 40 (див.табл.2), що відповідає динаміці курсу UAH в період з 7 жовтня 2008 р. по 2 грудня 2008 р.

Таблиця 2 – Навчальні дані для побудови моделі гібридної мережі

Перша вхідна змінна	Друга вхідна змінна	Третя вхідна змінна	Четверта вхідна змінна	Вихідна змінна
49.7824	50.2849	50.4827	50.0529	50.1502
50.1502	49.7824	50.2849	50.4827	48.6149
48.6149	50.1502	49.7824	50.2849	46.9711

46.9711	48.6149	50.1502	49.7824	45.9374
45.9374	46.9711	48.6149	50.1502	48.6234
48.6234	45.9374	46.9711	48.6149	50.7012
50.7012	48.6234	45.9374	46.9711	49.6897
49.6897	50.7012	48.6234	45.9374	50.8952
50.8952	49.6897	50.7012	48.6234	51.2519
51.2519	50.8952	49.6897	50.7012	49.9059
49.9059	51.2519	50.8952	49.6897	49.6307
49.6307	49.9059	51.2519	50.8952	48.0977
48.0977	49.6307	49.9059	51.2519	48.5072
48.5072	48.0977	49.6307	49.9059	46.9205
46.9205	48.5072	48.0977	49.6307	45.1746
45.1746	46.9205	48.5072	48.0977	45.9676
45.9676	45.1746	46.9205	48.5072	45.1269
45.1269	45.9676	45.1746	46.9205	43.3566
43.3566	45.1269	45.9676	45.1746	37.6230
37.6230	43.3566	45.1269	45.9676	45.2767
45.2767	37.6230	43.3566	45.1269	45.7421
45.7421	45.2767	37.6230	43.3566	46.2053
46.2053	45.7421	45.2767	37.6230	46.3590
46.3590	46.2053	45.7421	45.2767	46.5588
46.5588	46.3590	46.2053	45.7421	46.6907
46.6907	46.5588	46.3590	46.2053	47.3418
47.3418	46.6907	46.5588	46.3590	47.4856
47.4856	47.3418	46.6907	46.5588	47.8727
47.8727	47.4856	47.3418	46.6907	47.3806
47.3806	47.8727	47.4856	47.3418	47.5518
47.5518	47.3806	47.8727	47.4856	46.4917

46.4917	47.5518	47.3806	47.8727	45.8501
45.8501	46.4917	47.5518	47.3806	44.6502
44.6502	45.8501	46.4917	47.5518	44.1064
44.1064	44.6502	45.8501	46.4917	41.5959
41.5959	44.1064	44.6502	45.8501	41.5020
41.5020	41.5959	44.1064	44.6502	39.6468
39.6468	41.5020	41.5959	44.1064	40.0315
40.0315	39.6468	41.5020	41.5959	37.4319
37.4319	40.0315	39.6468	41.5020	37.1555

Збережемо навчальну вибірку в зовнішньому файлі під ім'ям priceUAH.dat. Після цього відкриємо редактор ANFIS, в який завантажимо цей файл з навчальними даними. Зовнішній вигляд редактора ANFIS із завантаженими навчальними даними представлений на рис.1.

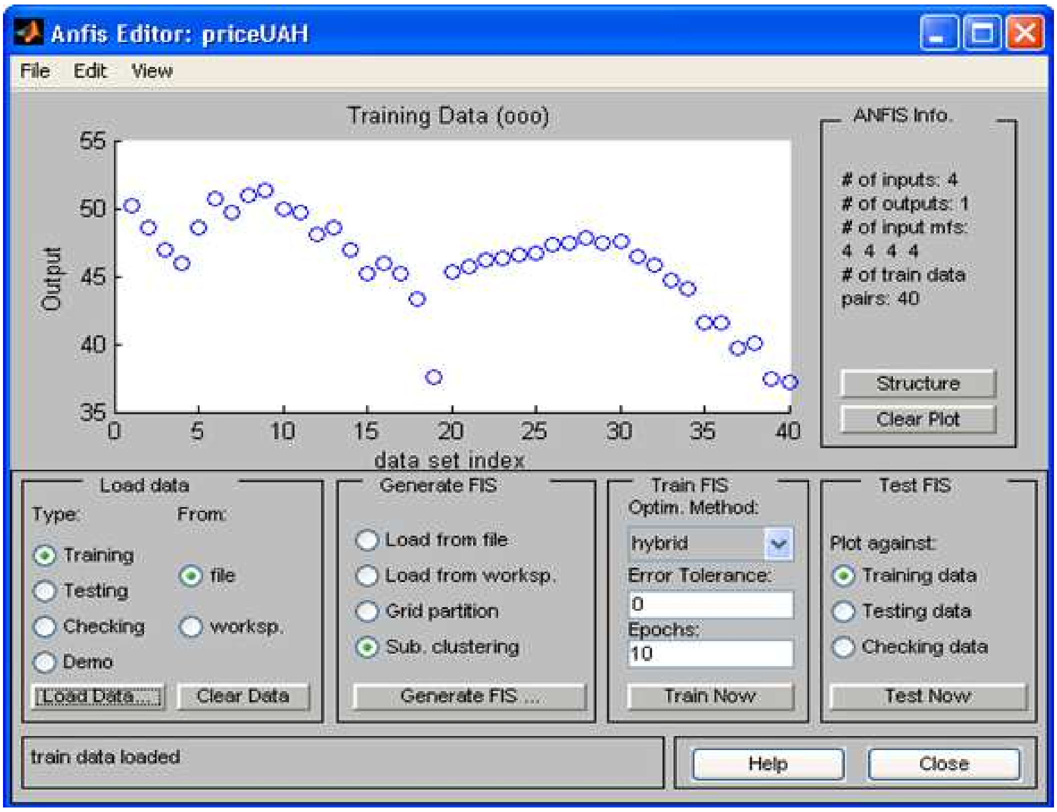


Рис.1 – Графічний інтерфейс редактора ANFIS після завантаження навчальних даних

Згенеруємо структуру системи нечіткого виведення типа Сугено за методом Sub.clustering. Для навчання гібридної мережі скористаємося гібридним методом навчання з рівнем помилки 0, а кількість циклів навчання задамо рівним 10. Після закінчення навчання даної гібридної мережі (за 2 ітерації) може бути виконаний аналіз графіку помилки навчання (рис.2).

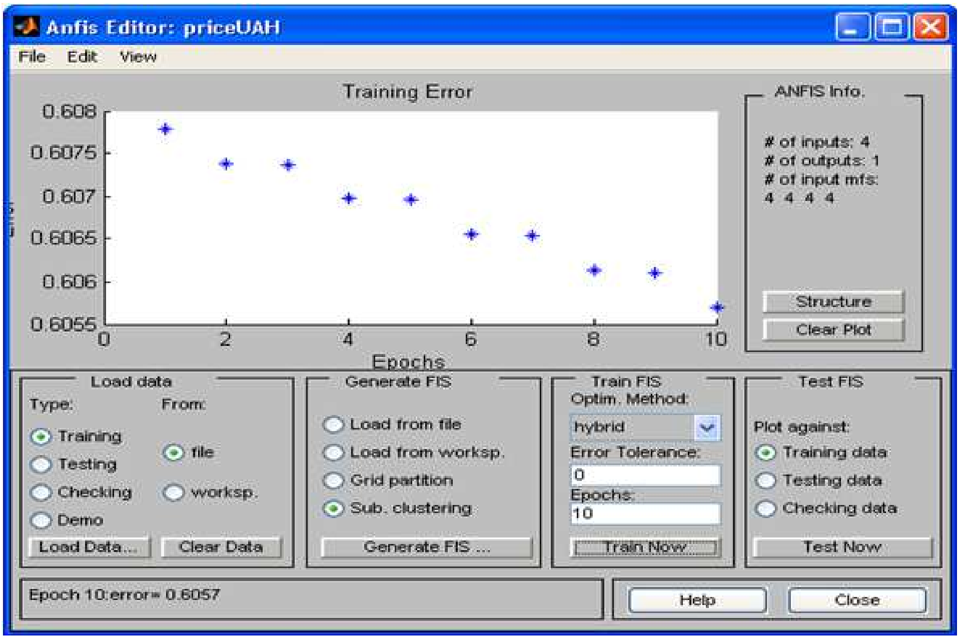


Рис.2 – Графік залежності похибки навчання від кількості циклів навчання

Після навчання гібридної мережі можна візуально оцінити структуру побудованої нечіткої моделі (рис.3).

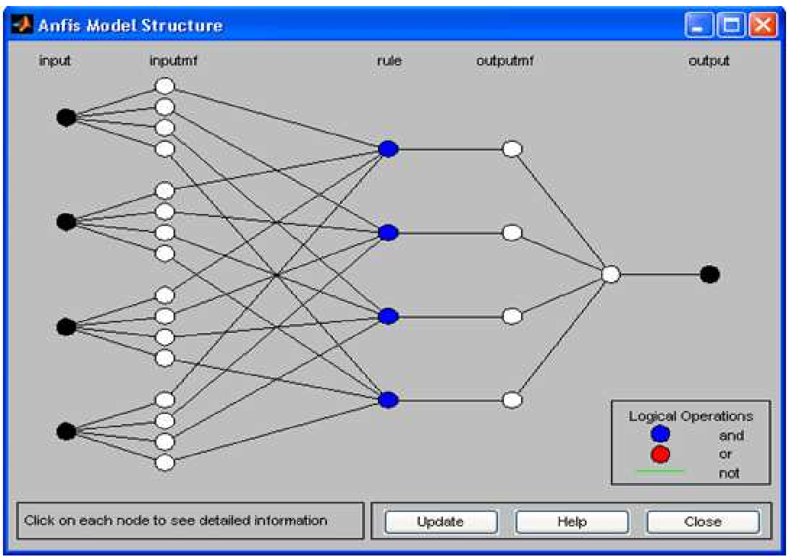


Рис.3 – Структура згенерованої системи нечіткого виведення

За допомогою графічних засобів системи MATLAB можна виконати контроль і налаштування параметрів функцій належності вхідних змінних і правил нечітких продукций. Для виконання відповідних операцій можна скористатися редактором функцій належності (рис.4). Проте до перевірки адекватності побудованої нечіткої моделі залишимо всі параметри функцій належності без змін.

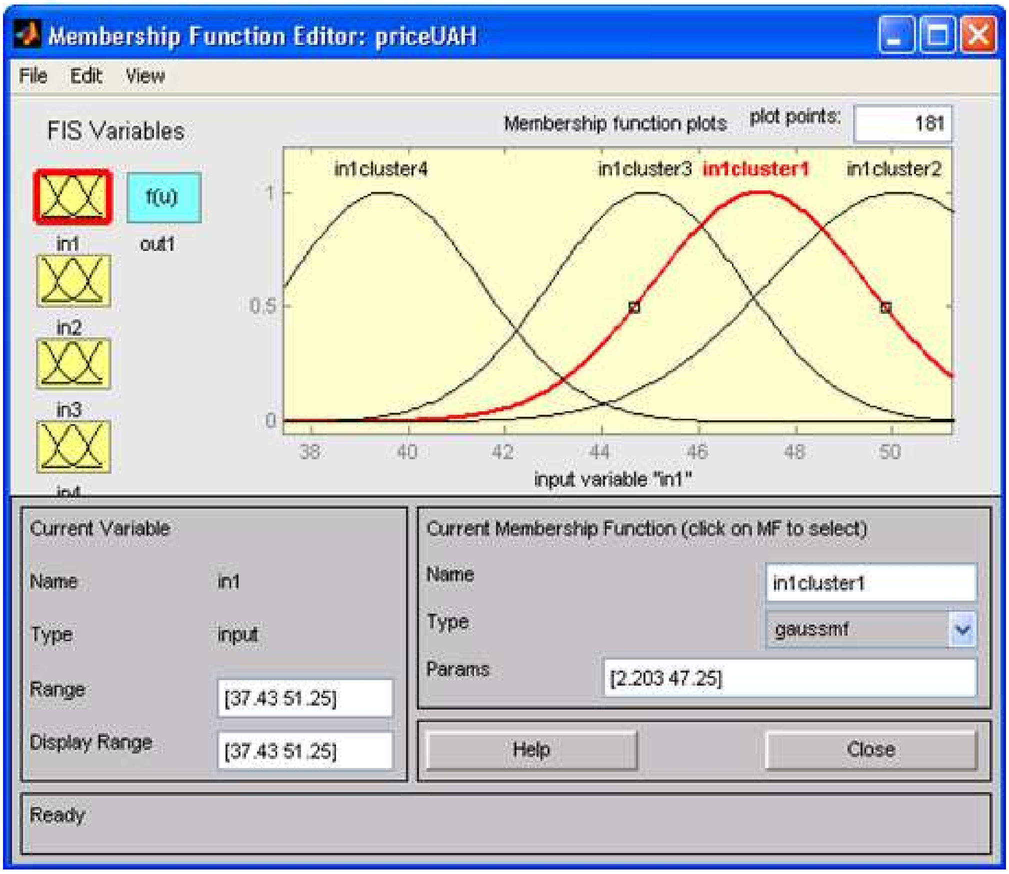


Рис.4 – Графічний інтерфейс редактора функцій належності розробленої системи нечіткого виведення для перевірки першої вхідної змінної

Виконаємо перевірку адекватності побудованої нечіткої моделі гібридної мережі. Для цієї мети зробимо ретроспективний прогноз значення курсової вартості UAH на наступний банківський день, наприклад, на 3 грудня 2008 р., вважаючи для цього випадку поточним банківським днем – 2 грудня 2008р.

Оскільки точність кількісних значень, що забезпечується графічними засобами пакету Fuzzy Logic Toolbox, є недостатньою для вирішення даного завдання, скористаємося функцією командного рядка evalfis. Як аргументи цієї функції вкажемо вектор значень курсової вартості UAH на поточний і 3 попередніх банківських дня. Повний формат виклику цієї функції буде наступним:

out = evalfis([37.1555 39.6468 41.5020 41.5959], priceUAH),

де out — умовне ім'я вихідної змінної; 37.1555 — значення курсової вартості UAH на 02.12.08; 39.6468 — значення курсової вартості UAH на 27.11.08; 41.5020 — значення курсової вартості UAH на 26.11.08; 41.5959— значення курсової вартості UAH на 25.11.08; priceUAH — ім'я структури FIS, заздалегідь завантаженої в робочу область системи MATLAB.

Після виконання цієї команди за допомогою розробленої нечіткої моделі буде набуто значення вихідний змінної для 3.12.08, рівне 37.7135 (рис.5).

```
>> out = evalfis([37.1555 39.6468 41.5020 41.5959], priceUAH)
Warning: Some input values are outside of the specified input range.
> In evalfis at 76

out =

   37.7135
```

Рис.5 – Вікно команд з відображенням результата оцінки розробленої нечіткої моделі гібридної мережі

Розглянутий підхід є перспективним напрямом для побудови і використання відповідних нечітких моделей прогнозування цін інших фінансових інструментів, таких як курси інших валют, акцій компаній, ф'ючерсів і опціонів. Дійсно, загальним для всіх цих інструментів з позицій технічного аналізу є відсутність апріорних припущень про динаміку коливань відповідних курсів цін, що сповна узгоджується з вихідними передумовами побудови нечітких моделей адаптивних систем нейро-нечеткого виведення.

ТЕМА 13. ПРОЕКТУВАННЯ ІЄРАРХІЧНИХ НЕЧІТКИХ СИСТЕМ

Нечітке виведення при нечітких вхідних даних

Розглянемо як розширити Fuzzy Logic Toolbox для виконання нечіткого виведення при нечітких вхідних даних. Передбачається, що функції належності нечітких вхідних даних задані гаусівською...

Нечітке логічне виведення для проміжних змінних з наступною передачею чітких значень цих змінних в нечіткі системи наступного рівня ієрархії

Для реалізації цього способу необхідно викликати функцію evalfis для кожної нечіткої бази знань. Розглянемо цей спосіб на прикладі реалізації нечіткої експертної системи прогнозування конкурентоспроможності марочного товару...

Нечітке логічне виведення для прямої передачі логічного виведення в вигляді нечіткої множини в машину нечіткого виведення наступного рівня ієрархії

Розглянемо як розширити можливості Fuzzy Logic Toolbox, щоб виконувати логічне виведення по ієрархічній нечіткій базі знань по другому способу, тобто без надлишкових дефазіфікації та фазіфікації проміжних змінних...

Нечітке виведення при нечітких вхідних даних

Розглянемо як розширити Fuzzy Logic Toolbox для виконання нечіткого виведення при нечітких вхідних даних. Передбачається, що функції належності нечітких вхідних даних задані гаусівською, колоколоподібною або сигмоїдною кривими.

При нечіткому моделюванні необхідно вміти визначати міри приналежності входів до термів із бази знань. Вони розраховуються по-різному при чітких і нечітких вхідних значеннях. В чіткому випадку міра приналежності розраховується підстановкою поточного значення змінної в формулу, наприклад, симетричної гаусівської функції приналежності:

$$\mu(x) = e^{-\frac{(x-h)^2}{2c^2}} \quad (1)$$

де c – коефіцієнт концентрації; h – координата максимуму. Ця функція в пакеті Matlab має ім'я gaussmf, порядок її параметрів: [c h].

При нечітких вхідних даних необхідно визначити міру приналежності однієї нечіткої множини – значення вхідної змінної, до іншої нечіткої множини – терму із бази знань. Міра приналежності дорівнює висоті перерізу цих нечітких множин (див. рис.1).

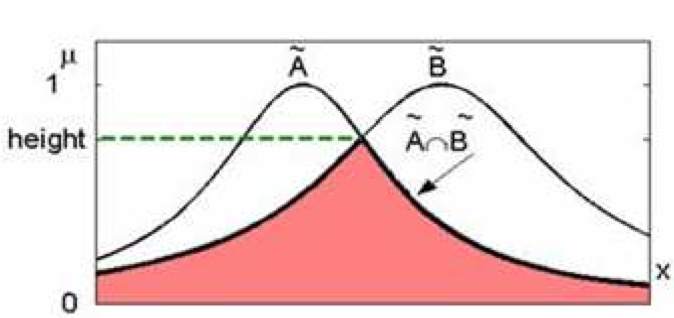


Рис.1 – Розрахунок міри приналежності нечіткої множини нечіткій множині

Функція fuzzy_input розраховує висоту пересічення двох нечітких множин. Функції належності нечітких множин мають бути одного типу. Допустимі типи функцій належності: колоколоподібна, гаусівська та сигмоїдна. Програму можна легко модифікувати під інші типи функцій належності. Лістинг програми fuzzy_input приведений нижче.

```
function t = fuzzy_input(param1, param2, mftype)
%FUZZY INPUT розраховує міру належності однієї
%нечіткої множини до іншої нечіткої множини:
%param1 і param2 - параметри функцій належності нечітких множин.
%Для 'gbellmf' коефіцієнти крутизни функцій належності
%мають бути однаковими: param1(2)== param2(2).
%mftype - тип функції належності. Допустимі типи
%функцій належності: 'gaussmf','sigmf' і 'gbellmf'.
```

```
%Нечіткі множини мають бути задані функціями
%належності одного типу.
%Потрібні ресурси: gaussmf.m, gbellmf.m i sigmf.m iз
%Fuzzy Logic Toolbox v2

switch lower (mftype) ;
 case 'gaussmf'
 sigma1 = paraml(1); c1 = param1(2);
 sigma2 = param2(1); c2 = param2(2);
if cl==c2 t=1;
else x1=(c1*sigma2 + c2*sigma1)/(sigma2 + sigma1);
 yl=gaussmf(x1, param1);
if sigma1==sigma2 y2=yl;
else x2=(cl*sigma2 - c2*sigma1)/(sigma2 - sigma1);
 y2=gaussmf(x2, paraml);
 end
 t = max(yl,y2);
end
case 'sigmf'
 a1=param1(1); c1 = param1(2);
 a2=param2(l); c2 = param2(2);
if ((al==a2)& (cl==c2)) t=1;
elseif al==a2 t=0;
else x=(a2*c2-al*cl)/(a2-al);
 t = sigmf(x,paraml);
end
case 'gbellmf'
if param1(2)~=param2(2) error('Для gbellmf коефіцієнти крутизни мають бути однаковими: param1{2)=param2(2)')
else
```

```
al=param1(1); cl=param1(3);
a2=param2(l); c2=param2(3);
if c1==c2 t=l;
else
x=(a2*cl+al*c2)/(al+a2);
y1 = gbellmf(x,param1);
if al == a2 y2=0;
else x=(a2*cl-al*c2)/(-a1+a2);
y2=gbellmf(x,paraml);
end
t = max(y1,y2);
end
end
otherwise
error {'Функція належності має бути gaussmf, sigmf або gbellmf')
end
```

При нечіткому виведенні за допомогою функції evalfis потрібно якимсь чином вказати, що вхідні змінні задані нечіткими числами. Для цього нечітке значення вхідної змінної кодується таким чином:

Fuzzy_value = FUZZY_LABEL-10*A-B,

де A – номер змінної в нечіткій системі; B – порядковий номер терма; FUZZY_LABEL – ознака нечіткого аргументу, позначена від'ємним числом.

Якщо FUZZY_LABEL = -1000, тоді аргументи $x \geq -1010$ розглядаються як чіткі, а аргументи $x < -1010$ – як нечіткі. Наприклад, якщо FUZZY_LABEL = -1000, тоді значення x=-1012 відповідає вхідній нечіткій множині з функцією належності другого терма першої вхідної змінної. Таким чином, всі нечіткі значення вхідних даних мають бути описані в нечіткій системі (fis-структурі). Ця fis-структура, а також константа FUZZY_LABEL мають бути доступні з m-функцій, в яких запрограмовані формули функцій належності. Це реалізується через глобальні змінні FUZZY_LABEL і FISFI (нечітка система).

Запрограмовано три функції належності, аргументами яких можуть бути як чіткі, так і нечіткі числа. Це функції:

• fsigmf – аналог сигмоїдної функції належності sigmf;

• fgaussmf – аналог гаусівської функції належності gaussmf;

• fgbellmf – аналог колоколоподібної функції належності gbellmf. Аби нечіткі системи могли працювати з нечіткими вхідними даними, необхідно в fis-файлах замінити імена функцій належності вхідних змінних з sigmf на fsigmf, з gaussmf на fgaussmf і з gbellmf на fgbellmf. Якщо використовуються функції приналежності інших типів, то необхідно написати аналогічні fgaussmf.m функції і вставити новий блок case у файлі fuzzy_input.m. Текст функції fgaussmf.m приведений нижче.

```
function y = fgaussmf(x, params)
%FGAUSSMF - гаусівська функція належності, за якою,
%на відміну від gaussmf можна розрахувати міру належності
%при нечіткому вхідному аргументі.
%Потрібні ресурси: gaussmf.m із Fuzzy Logic Toolbox v2
global FISFI %нечітка система
global FUZZY_LABEL %ознака нечіткого аргумента, задається від'ємним числом <= -100.
% Якщо FUZZY_LABEL = -1000,
%тоді аргументи x >= -1010 розглядаються як чіткі, а аргументи
% x < -1010 - як нечіткі.
%Нечітке значення кодується так: FUZZY_LABEL-10*A-B, де A - номер змінної в FISFI,
%B - номер терму.
y=gaussmf(x,params);
if max(max(x<FUZZY_LABEL-10))%Якщо є нечіткі значення
[ii jj]=size (x);
for i=1:ii
```

```
for j=1:jj
if x(i,j)<(FUZZY_LABEL-10) %Якщо нечітке значення на вході
 xx=x(i,j)-FUZZY_LABEL;
 var_index=floor(-.1*xx);
 term_index=-1*xx-var_index*10;
 mf_type=lower(FISFI.input(var_index)....
 mf(term_index).type);
if (strcmp(mf_type,'gaussmf')|strcmp(mf_type,'fgaussmf')
%Міра належності однієї нечіткої множини до іншої:
params2=FISFI.input(var_index)....
mf(term_index).params;
y(i,j)=fuzzy_input(params, params2,'gaussmf');
else error('Функція належності вхідного нечіткого числа має бути
%gaussmf або fgaussmf')
end
end
end
end
end
```

Перед викликом функції нечіткого виведення evalfis необхідно об'явити глобальні змінні FUZZY_LABEL і FISFI і привласнити їм відповідні значення. Значення FUZZY_LABEL встановлюється таким, аби воно було менше лівої межі діапазонів зміни вхідних змінних нечіткої системи FISFI.

Як приклад приведемо сценарій виконання нечіткого виведення при x_1 =Low, x_2 = Average і x_3 = Low для нечіткої системи 'sample fuzzy.fis':

```
global FISFI
global FUZZY_LABEL
FISFI = readfis(‘sample_fuzzy');
FUZZY_LABEL = -1000;
evalfis([-1011 -1022 -1031], FISFI)
```

В результаті отримуємо:

ans = 0,4006

Для вхідних значень x_1 = Low, x_2 = Average і x_3 = 0,2 наберемо:

evalfis([-1011 -1022 0.2], FISFI)

В результаті на екрані отримуємо:

Warning: Some input values are outside of the specified input range.

>In C:\MATlAB7\toolbox\fuzzy\fuzzy\evalfis.m at line 73.

ans = 0,5161

Для виведення без попередження можна замість функції evalfis використовувати функцію evalfis_ww.

Нечітке логічне виведення для проміжних змінних з наступною передачею чітких значень цих змінних в нечіткі системи наступного рівня ієрархії

Для реалізації цього способу необхідно викликати функцію evalfis для кожної нечіткої бази знань. Розглянемо цей спосіб на прикладі реалізації нечіткої експертної системи прогнозування конкурентоспроможності марочного товару.

Розглядається продукція підприємства, що продається під торгівельною маркою (брендом). Конкурентоспроможність є однією з важливих маркетингових характеристик товару. Вона виражає сукупну здатність товару витримати конкуренцію з іншими товарами на певному ринку, бути реалізованим і принести прибуток.

Критерієм конкурентоспроможності марочного товару називається число Q є [0,100] . Чим більше значення цього критерію, тим більше шансів у марочного товару бути вибраним покупцями, тим більше його сегмент ринку. На конкурентоспроможність марочного товару впливає багато виробничих, психоло-гічних, соціальних, політичних і інших чинників. Позначимо їх через x_1, x_2, …, x_n , тоді модель оцінки конкурентоспроможності продукції підприємства представлятиме функціональне відображення у такому вигляді:

$$V = \sum_{r=1,H} M_r ,$$

де X – вектор чинників, що впливають на конкурентоспроможність продукції.

При великій кількості чинників їх вплив зручно класифікувати в вигляді ієрархічного дерева логічного виведення (рис.1).

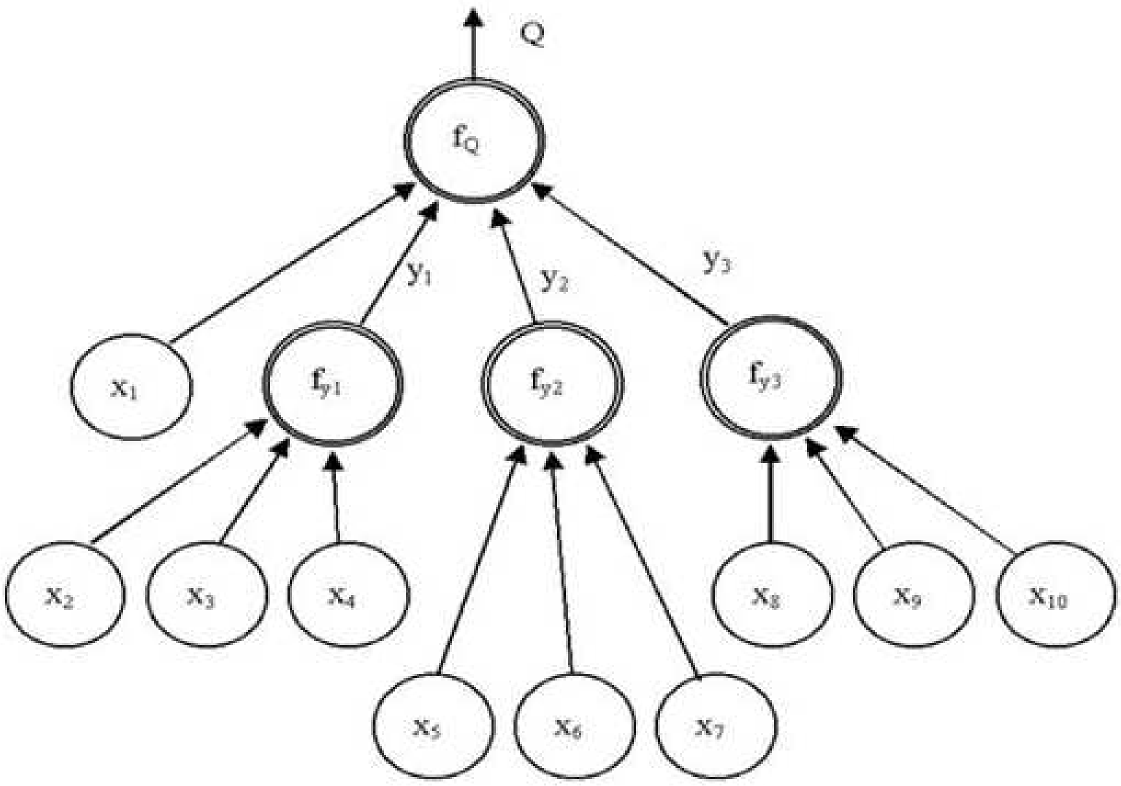


Рис.1 – Ієрархічна класифікація чинників, що впливають на конкурентоспроможність

Елементи дерева логічного виведення інтерпретуються наступним чином:

· корінь дерева – конкурентоспроможність марочного товару (Q);

· термінальні вершини – часткові чинники впливу (x_1, x_2, …, x_n);

· нетермінальні вершини (подвійні кола) – згортки впливаючих чинників впливу;

· дуги графа, що виходять з нетермінальних вершин – укрупнені чинники впливу (y_1, y_2, …, y_n).

Згортки fy1, fy2, fy3 виконуються за допомогою логічного виведення за нечіткими базами знань.

Опис чинників, що впливають на конкурентоспроможність марочного товару, наведений в табл.1. Значення чинників будемо виражати як відхилення (у відсотках) від усереднених показників по аналогічних товарах конкуруючих брендів на аналізованому ринку. Для моделювання укрупнених впливаючих чинників (ціни, якості, іміджу та сервісу) використовуються експертні нечіткі бази знань типа Мамдані, які наведені в табл..2-4. Елементи посилок (антецедентів) нечітких правил зв’язані логічною операцією ТА.

Таблиця 1 – Чинники, що впливають на конкурентоспроможність продукції

Найменування чинника	Опис чиннику
y_1 – якість	Сукупність споживчих властивостей; здатність задовольняти очікувані потреби споживача
y_2 – імідж	Цілісна сукупність асоціацій і вражень, що представляє торгівельну марку в свідомості споживача
y_3 – сервіс	Множина послуг, знижок і пільг, що надаються реальним і потенційним споживачам марочного товару
x_1 – ціна	Роздрібна ціна марочного товару на аналізованому ринку
x_2 – якість проектних рішень	Потенційна якість, закладена в марочний товар. Визначається: для харчових продуктів – рецептурою; для апаратури – схемотехнічними рішеннями; для одягу – дизайном; для освітніх послуг – учбовими планами

x_3 – якість виробничих технологій	Об'єктивні обмеження на рівень потенційної якості. Для товаровиробників вони обумовлені технологічним процесом, засобами праці (устаткування, інструменти) і предметами праці (комплектуючі, сировина, інгредієнти). Для учбових закладів ці обмеження обумовлені лабораторною базою і учбовим процесом (розклад, технології навчання і контролю знань і тому подібне)
x_4 – кадрове забезпечення	Суб'єктивні обмеження на рівень потенційної якості, обумовлені кваліфікацією, дисциплінованістю і мотивованістю персоналу
x_5 – ранг виробника	Міра довіри до виробника марочного товару, що зазвичай визначається державними органами сертифікації. Наприклад, споживачі більше довіряють товарам, вироблених згідно з ДСТУ або ISO, чим приватним підприємцям, що випускають продукцію за ТУ. Для вузів ранг визначається рівнем акредитації і статусом (національний університет, університет, інститут або філія)
x_6 – рекламне забезпечення	Інформація, що поширюється на користь виробника марочного товару. Складається з рекламного забезпечення всього бренду і конкретного марочного товару. Визначається ідентичностями бренду і марочного товару (ім'я, товарний знак, історія і тому подібне), а також обсягом і якістю прямої та прихованої реклами
x_7 – рівень рекламацій	Інформація, що поширюється не на користь виробника марочного товару. Складається з рекламацій на конкретний марочний товар і на весь бренд.

	Визначається кількістю і мірою претензій споживачів, рівнем поширення інформації про рекламації, а також контрпропагандою конкурентів
x_8 – зручність покупки	Легкість здійснення покупки, що визначається географічною і часовою доступністю точок продажу, а також сервісним обслуговуванням при придбанні товару (консультації, доставка і установка)
x_9 – сервіс при експлуатації	Зручності при експлуатації марочного товару, що визначаються можливістю модернізації товару, терміном гарантійного і післягарантійного обслуговування, географічною і часовою доступністю сервісних центрів і точок збуту витратних матеріалів
x_{10} – бонуси	Додаткові пільги, знижки і послуги, що стають доступними споживачам марочного товару

Таблиця 2 – Нечітка база знань для оцінки якості марочного товару

x_2 – якість проектних рішень	x_3 – якість виробничих технологій	x_4 – кадрове забезпечення	y_1 - якість
Висока	Висока	Високе	Висока
Висока	Висока	Середнє	Висока
Висока	Середня	Високе	Висока
Середня	Висока	Високе	Висока
Середня	Висока	Середнє	Висока
Низька	Низька	Низьке	Низька
Низька	Низька	Середнє	Низька
Низька	Середня	Низьке	Низька
Середня	Низька	Низьке	Низька
Середня	Низька	Середнє	Низька

Висока	Низька	Середнє	Середня
Висока	Середня	Низьке	Середня
Низька	Висока	Середнє	Середня
Низька	Середня	Високе	Середня
Середня	Висока	Низьке	Середня
Середня	Низька	Високе	Середня
Середня	Середня	Середнє	Середня

Таблиця 3 – Нечітка база знань для оцінки іміджу марочного товару

x_5 – ранг виробника	x_6 – рекламне забезпечення	x_7 – рівень рекламацій	y_2 – імідж
Високий	Високе	Середній	Високий
Високий	Середнє	Низький	Високий
Будь-який	Високе	Низький	Високий
Середній	Високе	Середній	Високий
Будь-який	Низьке	Високий	Низький
Низький	Низьке	Середній	Низький
Низький	Середнє	Високий	Низький
Середній	Низьке	Середній	Низький
Високий	Низьке	Середній	Середній
Високий	Середнє	Високий	Середній
Низький	Високе	Середній	Середній
Низький	Середнє	Низький	Середній
Середній	Високе	Високий	Середній
Середній	Низьке	Низький	Середній
Середній	Середнє	Середній	Середній

Таблиця 4 – Нечітка база знань для оцінки рівня сервісу марочного товару

x8 – зручність покупки	*x9 – сервіс при експлуатації*	*x10 – бонуси*	*y3 – сервіс*
Висока	*Високий*	*Високі*	*Високий*
Висока	*Високий*	*Середні*	*Високий*
Висока	*Середній*	*Високі*	*Високий*
Висока	*Середній*	*Середні*	*Високий*
Середня	*Високий*	*Високі*	*Високий*
Низька	*Низький*	*Низькі*	*Низький*
Низька	*Низький*	*Середні*	*Низький*
Низька	*Середній*	*Низькі*	*Низький*
Низька	*Середній*	*Середні*	*Низький*
Середня	*Низький*	*Низькі*	*Низький*
Висока	*Низький*	*Середні*	*Середній*
Висока	*Середній*	*Низькі*	*Середній*
Низька	Високий	Середні	Середній
Низька	Середній	Високі	Середній
Середня	Високий	Низькі	Середній
Середня	Низький	Високі	Середній
Середня	Середній	Середні	Середній

Графіки функцій приналежності нечітких термів "Низький" (Н), "Середній" (С) і "Високий" (В) представлені на рис.2.

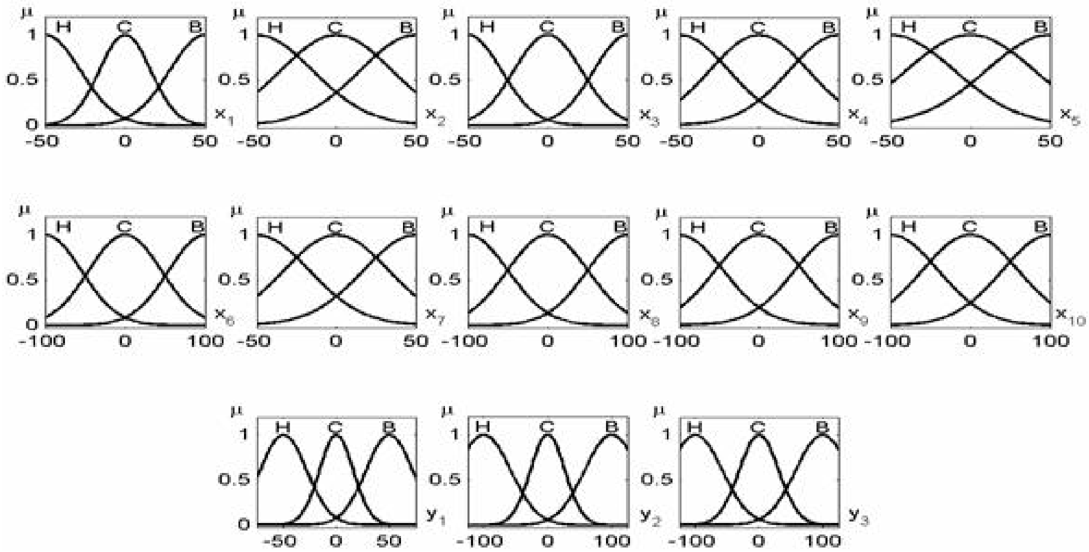


Рис.2 –Функції приналежності нечітких термів для прогнозування конкурентоспроможності

Конкурентоспроможність марочного товару необхідно моделювати з врахуванням трьох типів збуту, коли для споживача показники ціни ТА якості ТА іміджу ТА сервісу є: 1) поганими, 2) середніми і 3) добрими. Передбачається, що при кожному типові збуту еластичність конкурентоспроможності по чинниках постійна. Межі підобластей з постійними еластичностями конкурентоспроможності – нечіткі, що обумовлене плавним переходом одного типа збуту в іншій. У табл.5 представлена нечітка база знань типу Сугено для моделювання конкурентоспроможності марочного товару.

Таблиця 5 – Нечітка база знань для оцінки конкурентоспроможності марочного товару

Тип збуту	x_1 – ціна	y_1 – якість	y_2 – імідж	y_3 – сервіс	Конкуренто-спроможність, Q
Поганий	Висока	Низька	Низький	Низький	
Середній	Середня	Середня	Середній	Середній	

Добрий	Низька	Висока	Високий	Високий	

Кожне правило цієї бази знань моделює один тип збуту. Коефіцієнти у висновках правил задають чутливість конкурентоспроможності по відповідних чинниках. Коефіцієнти у висновках кожного правила або розраховуються по методу парних порівнянь Сааті або настроюються з використанням ANFIS-алгоритму. Для автоматизації розрахунків за методом аналізу ієрархій можна використати наступні m-файл-функції:

a2mu_r.m – розраховує вагові коефіцієнти із матриці парних порівнянь за методом Ротштейна:

```
function [norm_mu, mu]=a2mu_r(A)
%Розрахунок вагових коефіцієнтів із матриці парних порівнянь за методом
%Ротштейна А.П.
%mu – вагові коефіцієнти
%norm_mu – нормалізовані вагові коефіцієнти

mu=1./sum(A);
if (max(mu)>=1 | min(mu)<=0)
 error ('Матриця парних порівнянь неправильно сформована')
end
%Нормуємо міри принадежності:
norm_mu=mu/max(mu);
```

string2a_r.m – формує матрицю парних порівнянь за одним останнім рядком матриці парних порівнянь, в якій порівнюється один елемент з усіма іншими (включая і себе). Розрахунок елементів матриці здійснюється за методом Ротштейна:

```
function [A, warning]=string2a_r(comp_str)
%Розрахунок матриці парних порівнянь (А) по одному рядку (comp_str),
%в якому порівнюють один елемент з усіма (включая і себе).
```

Порівняння

%виконують за шкалою Сааті, тому допустимими значеннями являються:

%1/9, 1/8,...,1,2,....,9.

%Розрахунок виконується за методом, що запропонований Ротштейном А.П.

```
%Передбачається, що матриця парних порівнянь крім властивості
%діагональності та оберненої симетричності, має ще і властивість
%транзитивності.
%warning - попередження:
% 0 - все в порядку;
% -1 - матриця містить елементи <1/9;
% 1 - матриця містить елементи >9.

warning=0;
%Перевірка за допустимими значеннями парних порівнянь:
if (max(comp_str)<=9) & (min(comp_str)>=1/9)
 N=length(comp_str);
 if N<2 error ('Рядок парних порівнянь повинен мати хоча б 2 елементи')
 end
 %Вихідна матриця парних порівнянь:
 A=ones(N,N);
 tmp1=find(comp_str==1);
 if length(tmp1)==0 error ("Рядок парних порівнянь повинен мати 1')
 end
 %Номер базового елементу - порядковий номер елемента, з яким
 %порівнюються інші:
 k=tmp1(1);
 %Формуємо k-ий рядок матриці парних порівнянь
 A(k,:)=comp_str;
```

%Розраховуємо всі елементи матриці парних порівнянь на основі

%властивості транзитивності:

for i=1:N

A(i,:)=A(k,:)/A(k,i);

end

%Перевірка на допустимі значення елементів матриці парних порівнянь

if max(max(A))>9 disp('Матриця парних порівнянь містить елементи >9.'); warning=1;

elseif min(min(A))<1/9 disp('Матриця парних порівнянь містить елементи <1/9.'); warning=-1;

end

else error ('Парні порівняння повинні бути від 1/9 до 9.')

end

string2mu_r.m – розраховує за методом Ротштейна А.П. вагові коефіцієнти за одним останнім рядком матриці парних порівнянь:

function [norm_mu, mu, warning]=string2mu_r(comp_str)

%Розрахунок мір приналежності (mu) по одному рядку (comp_str) матриці

%парних порівнянь, в якому порівнюють один елемент з усіма (включая і себе).

%Порівняння виконують за шкалою Сааті, тому допустимими значеннями

%являються: 1/9, 1/8,...,1,2,....,9.

%Розрахунок виконується за методом, що запропонований Ротштейном А.П.

%mu – вагові коефіцієнти

%norm_mu – нормалізовані вагові коефіцієнти.

%warning - попередження:

% 0 - все в порядку;

% -1 - матриця містить елементи <1/9;

% 1 - матриця містить елементи >9.

%Знаходимо матрицю парних порівнянь:

[A, warning]=string2a_r(comp_str);

%Розраховуємо вагові коефіцієнти:

[norm_mu, mu]=a2mu_r(A);

Синтаксис визова описаних m-файл-функцій такий:

% За дев'ятибальною шкалою Сааті заповнюємо останній рядок матриці порівнянь

comp_str = [9 7 5 3 1 1];

% Формуємо матрицю парних порівнянь за методом Ротштейна А.П.

A=string2a_r(comp_str);

% Розраховуємо вагові коефіцієнти із матриці парних порівнянь

% mu – вагові коефіцієнти; norm_mu – нормовані вагові коефіцієнти

[norm_mu, mu] = a2mu_r(A)

Якщо інформація про матрицю парних порівнянь не потрібна, то розрахувати вагові коефіцієнти можна такою командою:

[norm_mu, mu] = string2mu_r(comp_str)

Нечітке виведення здійснюється алгоритмами Мамдані та Сугено. Як *t*-норма обраний добуток. При нечіткому методі Мамдані дефазіфікація здійснюється за методом центру тяжіння, а при виведенні Сугено – за методом зваженого середнього.

Ієрархічна нечітка модель оцінки конкурентоспроможності реалізована чотирма системами нечіткого виведення:

hy1.fis – нечітка система моделювання якості марочного товару (y_1);

hy2.fis – нечітка система моделювання іміджу марочного товару (y_2);

hy3.fis – нечітка система моделювання сервісу, асоційованого з марочним товаром (y_3);

hQ.fis – нечітка система прогнозування конкурентоспроможності марочного товару (Q).

Ієрархічне нечітке виведення по дереву виконує функція conc.m.

Ця функція повертає два вихідних аргументи: перший аргумент – результат оцінки конкурентоспроможності продукції підприємства; другий аргумент – значення укрупнених чинників впливу (y_1, y_2, y_3). Функція викликається з 10 вхідними аргументами, які задають значення факторів x_1 ¸x_{10} . Значення факторів x_2 ¸x_{10} можна задавати як числами, так і термами: “Н” – низький; “НС” – нижче середнього; “С” – середній; “ВС” – вище середнього; “В” – високий.

Для логічного виведення при нечітких вхідних даних використовується функція qgaussmf, модифікована з бібліотечної гаусівської функції приналежності gaussmf**.** Розрахунок міри приналежності однієї нечіткої множини до іншої нечіткої множини здійснюється функцією qual_inp_gauss. Логічне виведення здійснюється через функцію evalfis_ww, яка аналогічна бібліотечній функції evalfis, але не видає попередження при використанні нечітких вхідних даних.

Розглянемо використання нечіткої експертної системи на наступних прикладах. Показники деякого марочного товару А на регіональному ринку експерти оцінили так: x_1 =10% ; x_2 – “висока”, x_3 – “середня”; x_4 – “середнє”; x_5 – “середній”; x_6 – -50%; x_7 = -40%; x_8 = -30%; x_9 – “середній”; x_{10} = -80% .

Для розрахунку конкурентоспроможності необхідно виконати команду

P=conc(10,'В','С','С','С',-50,-40,-30,'С',-80)

В результаті отримаємо, що конкурентоспроможність цього марочного товару А є середньою з чисельним значенням Q_P =51,75. Виконання нечітких виведень Мамдані для цього прикладу ілюструє рис.3.

В результаті моделювання отримуємо, що конкурентоспроможність товару *Б:* Q_N =62,41. Конкурентоспроможність товару *А* дорівнює Q_P =51,75. Отже, в даному регіоні ринкові сегменти цих товарів знаходяться в співвідношенні: b_N =Q_N :Q_P =62,41:51,75=1,21:1.

Передбачимо, що для підйому конкурентоспроможності товару *А* до рівня = 63 (тобто більшого, ніж у конкурента) менеджер може змінювати ціну товару та рівень реклами в таких діапазонах: x_1Î[-20, 20] і x_6Î[-55, 10]. Застосовуючи запропоновану нечітку експертну систему, побудуємо графік залежності конкурентоспроможності товару *A* від чинників x_1 і x_6 (рис.3). Забезпечити бажану конкурентоспроможність = 63 можна зниженням ціни і збільшенням рівня реклами. Можливі варіанти заданого збільшення конкурентоспроможності показані на рис.5 ізолінією з маркером «63».

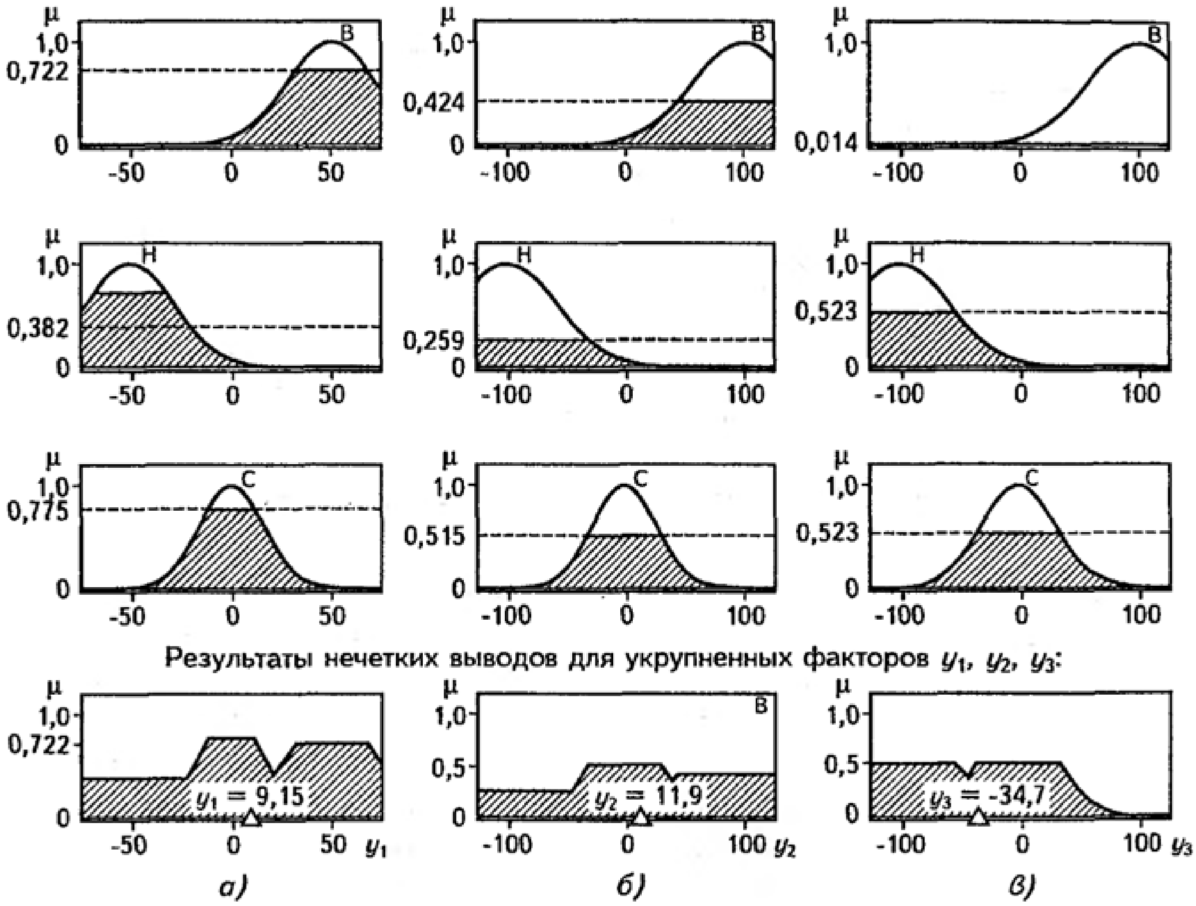


Рис. 3– Прогнозування укрупнених факторів y_1, y_2 і y_3 для товару А

а – якість; б – імідж; в – сервіс

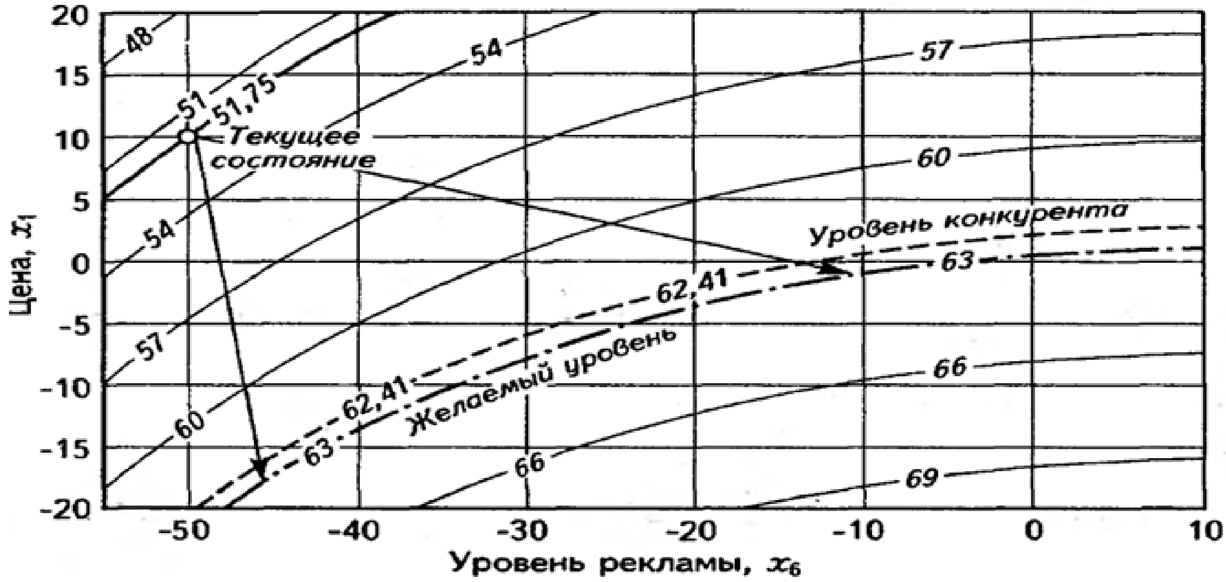


Рис.4 – Ізолінії конкурентоспроможності товару А

Модель конкурентоспроможності марочного товару побудована на основі лише експертних знань, тому можливі неспівпадання результатів нечіткого виведення (теорія) з експериментальними даними. Для забезпечення достовірних результатів необхідно провести параметричну ідентифікацію нечіткої моделі за експериментальними даними маркетингових досліджень. У нечіткій моделі настроюють параметри функцій приналежності термів, а також коефіцієнти у висновках правил в базі знань Сугено.

Для навчання нечіткої моделі експериментальні дані маркетингових досліджень представимо так:

$$(x_{rs}, \beta_{rs}),\ r=\overline{1,H},\ s=\overline{1,M_r}, \qquad (2)$$

де *H* - кількість ринків, на яких досліджуються конкурентоспроможності марочних товарів;

- кількість марочних товарів одного типа на *r*-ому регіональному ринку;

- вектор значень впливаючих чинників для s-го марочного товару на r-му регіональному ринку;

- доля r-го ринку, що припадає на s-ий марочний товар.

Для математичної постановки задачі навчання нечіткої моделі введемо такі позначення:

T - загальна кількість термів в нечітких базах знань;

c_l – коефіцієнт концентрації функції приналежності (1) l-го нечіткого терму ;

$C=(c_1, c_2, \ldots, c_T)$ – вектор коефіцієнтів концентрацій функцій приналежності;

$B=(b_{10}, b_{11}, b_{12}, b_{13}, b_{14}, b_{20}, b_{21}, b_{22}, b_{23}, b_{24}, b_{30}, b_{31}, b_{32}, b_{33}, b_{34})$ – вектор коефіцієнтів в висновках правил нечіткої бази знань Сугено;

$$V = \sum_{r=\overline{1,H}} M_r$$

V - обсяг навчальної вибірки (2)

Згідно теорії нечіткої ідентифікації навчання нечіткої моделі оцінки конкурентоспроможності продукції полягає у знаходженні такого вектора (B, C), щоб

$$RMSE = \sqrt{\frac{1}{V} \sum_{r=\overline{1,H}} \sum_{s=\overline{1,M_r}} \left(\beta_{rs} - \beta_{rs}^F\right)^2} \to \min \quad , (3)$$

де β_{rs}^F – спрогнозована за нечіткою моделлю з параметрами (B, C) доля r-го ринку, що приходиться на марочний товар з показниками Xrs. Для розрахунку β_{rs}^F необхідно за нечіткою моделлю знайти конкурентоспроможності , , …, присутніх на r-му ринку марочних товарів і використати формулу

$$\beta_{rs}^{F}=\frac{Q_{rs}\cdot 100\%}{Q_{r1}+Q_{r2}+\ldots+Q_{rM_r}}.$$

Комп'ютерні експерименти показують, що добрі результати оптимізації отримуються, коли обсяг навчальної вибірки в 2 і більше разів перевищує кількість параметрів, що настроюються. Тому, для навчання нечіткої моделі необхідно близько 80 пар експериментальних даних "входи-вихід". При меншому обсязі навчальної вибірки (30£S£80), краще настроювати тільки лінійні параметри нечіткої моделі, що представлені вектором *B*.

Нечітке логічне виведення для прямої передачі логічного виведення в вигляді нечіткої множини в машину нечіткого виведення наступного рівня ієрархії

Розглянемо як розширити можливості Fuzzy Logic Toolbox, щоб виконувати логічне виведення по ієрархічній нечіткій базі знань по другому способу, тобто без надлишкових дефазіфікації та фазіфікації проміжних змінних. Для цього необхідно виконати програмування двох процедур. Перша процедура повинна видавати результат виведення по проміжний базі знань в вигляді нечіткої множини типа:

$$\tilde{y}=\left(\frac{\mu_{d_1}(y)}{d_1},\frac{\mu_{d_2}(y)}{d_2},\ldots,\frac{\mu_{d_m}(y)}{d_m}\right),$$

де Mdj(y) – міра приналежності результата нечіткому терму d_j , ; m – кількість термів. Друга процедура повинна передавати знайдені міри приналежності Mdj(y), в машину нечіткого виведення наступного рівня ієрархії.

Для першої процедури використаємо прийоми нечіткої класифікації. Як проміжну нечітку систему використаємо систему нечіткого виведення Сугено (в Fuzzy Logic перетоворення з нечіткої системи Мамдані в нечітку систему Сугено здійснюється за допомогою функції mam2sug). Класам рішень поставимо у відповідність "функції належності" вихідної змінної. Параметри висновків правил (параметри "функцій належності" вихідної змінної) можуть бути довільними, бо вони не впливають на результуючу нечітку множину. Для отримання результатів логічного виведення в вигляді нечіткої множини потрібно виконати наступні команди:

```
%Нечітке виведення:
[a,b,c,d]=evalfis(inputs,fis); %inputs - входи, fis - нечітка система
%Визначення мір належності:
number_of_terms=length(fis.output(1).mf); %кількість термів
mf_grades=zeros(1, number_of_terms); %шукані міри належності
number_of_rules=length(fis.rule); %кількість правил в базі знань
for j=1:number_of_rules
term_index=fis.rule(j).consequent;
%s-норма, що реалізована максимумом
mf_grades(term_index)=max(mf_grades(term_index), d(j));
end
```

Для реалізації другої процедури пропонується спеціальна функція належності treemf, яка повертає міру належності, що дорівнює її першому параметру:

```
function mu = treemf(x, params)
mu=x; %Щоб не визначати розмір
mu(:,:)=params(1);
```

Цей тип функції належності слід встановити для кожного терму вхідної проміжної змінної. Для передачі в машину нечіткого логічного виведення наступного рівня ієрархії мір належності, знайдених першою процедурою, необхідно перший параметр функції належності treemf встановити рівним відповідній координаті вектора mf_grades. Параметри функцій належності вхідних проміжних змінних необхідно встановлювати при кожному нечіткому виведенні.

Для автоматизації логічного виведення по ієрархічній базі знань можна використати наступні m-файл-функції: treemf, prepare_tree, hier_evalfis.

Якщо для дерева логічного виведення, представленого на рис.14.2, нечіткі системи типу Сугено y1=f1(x2,x3,x4), y2=f2(x5,x6,x7), y3=f3(x8,x9,x10) і Q=f4(x1,y1,y2,y3) записані в файлах hy1.fis, hy2.fis, hy3.fis і hQ.fis, то синтаксис виклику описаних m-файл-функцій такий:

```
% список fis-файлов:
fis_file_names ={'hy1.fis' 'hy2.fis' 'hy3.fis' 'hQ.fis'};
%матриця зв'язків між базами знань:
tree_list=[4 2; 4 3; 4 4];
[FIS_list, input_list]=prepare_tree(fis_file_names, tree_list)
%Значення вхідних (термінальних) змінних
X1=[-10 20 10 50 -20 50 40 -30 -20 30]
out1=hier_evalfis(X1, FIS_list, input_list, tree_list, 1)
```

ТЕМА 14. НЕЧІТКА ІЄРАРХІЧНА МОДЕЛЬ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ

Фазіфікація вербальних оцінок експерта за шкалою відносної важливості

Майже кожна більш-менш складна економічна задача прийняття рішення є задачею прийняття рішень в умовах ризику за наявності багатьох критеріїв. Задача оцінки скінченної множини варіантів (стратегій) і векторної оптимізації пов’язана з невизначеністю при спробі виявити взаємну (відносну) важливість різних критеріїв щодо прийняття рішень...

Формування багаторівневої ієрархічної структури критеріїв

На верхньому рівні цієї структури (рівень 0) знаходиться лише один елемент — інтегрований критерій оцінювання, який можна розкласти (деталізувати) на кілька елементів (часткових критеріїв), тобто рівень...

Побудова матриці попарних порівнянь елементів з нечіткими оцінками

Для аналізу критеріїв оцінювання, що містяться на певних рівнях ієрархічної структури, пропонується побудова матриці попарних порівнянь елементів у вигляді...

Обчислення векторів нечітких ваг елементів ієрархічної структури

Для наведеної на рис.16.1 ієрархічної структури знаходимо, наприклад, вагові коефіцієнти критеріїв...

Впорядкування досліджуваних об’єктів щодо величини нечітких оцінок

Таке впорядкування не можна виконати коректно, якщо спиратися лише на максимальні величини носіїв нечітких множин...

Фазіфікація вербальних оцінок експерта за шкалою відносної важливості

Майже кожна більш-менш складна економічна задача прийняття рішення є задачею прийняття рішень в умовах ризику за наявності багатьох критеріїв. Задача оцінки скінченної множини варіантів (стратегій) і векторної оптимізації пов'язана з невизначеністю при спробі виявити взаємну (відносну) важливість різних критеріїв щодо прийняття рішень.

При **побудові нечітких моделей** важливу роль відіграють **лінгвістичні змінні**. З їх допомогою за певними правилами можна формалізувати (девербалізувати) якісну інформацію стосовно об'єкта прийняття рішень, представлену у словесній формі фахівцями-експертами, використовуючи нечіткі множини.

У наукових публікаціях щодо прийняття рішень сполучення термінів «багатокритеріальний» і «нечіткий» зустрічаються досить часто. Однак більшість авторів, сформулювавши на початку аналізу задачу прийняття рішень як нечітку та багатокритеріальну, вже на першому ж кроці її розв'язання використовують певну згортку критеріїв і надалі вивчають скалярну нечітку задачу прийняття рішень. Згортки, як правило, вводяться інтуїтивно, на підставі здорового глузду, залежно від змісту конкретної задачі, раціональних суджень тощо.

Вітлінським В.В. запропонований дещо інший підхід – **модифікація методу аналізу ієрархій** (МАІ), розробленого *Сааті* для підтримки прийняття рішень при багатокритеріальному виборі одного варіанту з множини варіантів. У літературі описані алгоритми, які для подолання браку кількісних даних використовують інформацію (оцінки), отриману на підставі спеціально прийнятих і використовуваних штучних шкал оцінювання. Втім, застосування такого підходу, що будується на описовому характері даних, не завжди дає бажані результати. Запропонована Вітлінським В.В. модифікація МАІ полягає у використанні понятійного і математичного апаратів теорії нечітких множин.

Нехай X – довільна непуста множина. Нечіткою підмножиною множини X називається множина пар

$$\widetilde{A} = \{(\mu_{\widetilde{A}}(x)/x)\}, \text{ де } x \in X,\ \ \mu_{\widetilde{A}}(x) \in [0;\, 1]. \qquad (1)$$

Функція

$$\mu_{\widetilde{A}}(x): \mu \in [0;\, 1]$$

називається **функцією належності до нечіткої множини** А, а X — базовою множиною чи базовою шкалою. Для кожного конкретного значення x є X величина $\mu_{\widetilde{A}}(x)$ набуває певного значення із замкненого інтервалу [0;1], що називається ступенем належності елемента x до нечіткої множини А. Носієм нечіткої множини А називається підмножина множини X, яка містить лише ті елементи множини X, в яких значення функції належності $\mu_{\widetilde{A}}(x)$ (2) є більшими від нуля. Функції належності однієї й тієї ж множини можуть бути різними при визначенні їх як різними людьми, так і однією людиною залежно від настрою останньої, схильності до ризику, від мети побудови нечіткої підмножини, розв'язуваної задачі, обраної конкретної методики побудови тощо.

Лінгвістична змінна характеризується набором (b, T(b), X, G, M), в якому b — назва лінгвістичної змінної; T(b) — терм-множина лінгвістичної змінної b, тобто множина лінгвістичних (вербальних) значень змінної, кожне з яких є нечіткою змінною з областю визначення X; G — синтаксичне правило (має звичайну форму граматики), що породжує назви (імена) a Î T(b) вербальних значень лінгвістичної змінної b; M — семантичне правило, яке ставить у відповідність кожній нечіткій змінній aÎT(b) нечітку множину С(а)— зміст нечіткої змінної a. Для спрощення запису формул позначають множину С(а)як с, а множину T(b) як T, коли йдеться про певні нечіткі змінні a і лінгвістичні змінні b. Окрім цього, використовують спрощене визначення лінгвістичної змінної як трійки (b, T, X), вкладаючи у позначення те саме розуміння, що й вище.

Нагадаємо, що лінгвістичною змінною є змінна, яка задана на деякій шкалі і приймає значення, що є словами та словосполуками природної чи штучної мов. Значення лінгвістичної змінної описуються нечіткими множинами.

Ступінь належності $\mu_{\tilde{A}}(x)$ елементів x до нечіткої множини інтерпретується як суб'єктивна міра того, наскільки елемент $x \in X$ відповідає поняттю, сутність якого формалізується нечіткою множиною. Під суб'єктивною мірою розуміють, як правило, визначений опитуванням експертів ступінь відповідності елемента $x \in X$ поняттю, яке формалізується нечіткою множиною.

Існує два класи методів побудови функції належності нечіткої множини — *прямі* та *непрямі* (*опосередковані*). Найпростіше функція належності нечіткої множини, що відповідає значенню лінгвістичної змінної *b*, будується за **прямими** методами для одного експерта. При побудові експерт кожному елементу множини *X* ставить у відповідність певний ступінь належності (2), який, на його погляд, найкращим чином узгоджується зі змістовним навантаженням (інтерпретацією) множини . Відповідність між ступенями належності з інтервалу [0; 1] та елементами *x* множини *X* може бути задана у вигляді таблиці, графіка, формули, що задає аналітичну форму функції належності нечіткої множини $\tilde{A}$ $(\tilde{A} \subset X)$

Непрямі методи ґрунтуються на «обережнішому» використанні особи як вимірювального приладу. Найбільш застосовуваним з цієї групи є метод попарних порівнянь. Розглянемо його сутність.

Функція належності Ма визначається, зокрема, за матрицею попарних порівнянь M=|mij|, елементи mij якої являють собою деякі оцінки інтенсивності належності елементів *xi є X* до нечіткої множини А у порівнянні з елементами xj є X. Якщо припустити, що значення функції належності Ма відомі для всіх *xi є X*, наприклад, Ма (*xi* = {1,2,...,*n*}), то попарні порівняння можна представити квадратною матрицею відношень *M*

= $\{m_{ij}\}$, де $m_{ij} = r_i / r_j$. Якщо відношення точні, то маємо співвідношення:

$$M \cdot r' = nr', \qquad r = (r_1, r_2, \ldots r_n),$$

де n — власне значення матриці M, знаючи яке, можна відшукати вектор r, r' — *вектор-стовпчик, транспонований до r.*

В загальному випадку емпіричний вектор r повинен задовольняти задачу знаходження власного значення матриці М, де — найбільше власне значення. Задача зводиться до знаходження власного вектора r, який задовольняє рівнянню:

$$M \cdot r' = \lambda_{\max} r' . \qquad (3)$$

Оскільки відомо, що це рівняння має єдиний розв'язок, то значення координат власного вектора r, який відповідає максимальному власному значенню , поділені на їх суму, будуть шуканими ступенями належності.

Щодо одержання матриці попарних порівнянь, то вона будується таким чином. Проводиться опитування експерта відносно того, наскільки, на його думку, елемент є більш значущий для поняття, що описується нечіткою множиною, ніж елемент . Поняття, якими може оперувати експерт, та детерміністична інтерпретація цих понять відповідними величинами наведені у табл.1.

Припустимо, що опитування експертів проведено на детерміністських засадах бездоганно і матриця попарних порівнянь побудована абсолютно точно. Тоді матриця М має такий вигляд:

Таблиця 1 – Шкала відносної важливості

Інтенсивність (вага) відносної важливості	Якісна оцінка (терм лінгвістичної змінної)	Пояснення
1	Однаково важливі	Обидва елементи роблять однаковий внесок щодо досягнення кінцевої цілі
3	Не набагато важливіший	Існують висловлювання відносно пріоритету одного елемента щодо іншого, але ці висловлювання досить непереконливі
5	Суттєво важливіший	Існують достатньо переконливі докази та логічні критерії, що один з елементів є важливішим (вагомішим)
7	Значно важливіший	Існують переконливі докази великої значущості одного елемента порівняно з іншим
9	Абсолютно важливіший	Усвідомлення пріоритету одного елемента щодо іншого максимально підтверджується
2, 4, 6, 8	Проміжні оцінки між двома сусідніми судженнями	Потрібен певний компроміс
$\frac{1}{v}; v = 1,...,9$	Обернені значення ненульових оцінок	Якщо елементу j1 при порівнянні з елементом j2 надається одна з ненульових інтенсивностей, то елементу j2 при порівнянні з j1 надається обернене значення цієї інтенсивності
0	Непорівняльність	Немає сенсу в порівнюванні елементів

$$M = \begin{vmatrix} \frac{r_1}{r_1} & \frac{r_1}{r_2} & \dots & \frac{r_1}{r_n} \\ \frac{r_2}{r_1} & \frac{r_2}{r_2} & \dots & \frac{r_2}{r_n} \\ \dots & \dots & \dots & \dots \\ \frac{r_n}{r_1} & \frac{r_n}{r_2} & \dots & \frac{r_n}{r_n} \end{vmatrix}$$

У такому разі для визначення j-го елемента вектора r можна скористатися такою процедурою. Обчислимо суму елементів j-го стовпчика матриці M. Нехай ця сума є деяке число Kj, тобто:

$$\sum_{i=1}^{n} m_{ij} = k_j, \; j = 1, \dots, n.$$

Одержимо, що $$\sum_{i=1}^{n} m_{ij} = \frac{\sum_{i=1}^{n} r_i}{r_j} = \frac{1}{r_j}.$$

Здійснюючи процедуру, аналогічну попередній, по всіх стовпчиках матриці *M*, будуємо шуканий вектор *r*.

Тепер припустимо, що, як це часто має місце, матриця попарних порівнянь побудована неточно. Тоді описану процедуру щодо визначення вектора *r* можна використати для визначення лише його початкового наближеного значення в ітераційному процесі розв'язку рівняння (3). При цьому відхилення λ_{max} від *n* може бути використане для оцінки точності розв'язку системи рівнянь на певному кроці ітераційного методу.

Запропонований Вітлінським В.В. алгоритм є однією з модифікацій **МАІ** і складається з п'яти основних кроків. Їх черговість і спектр основних операцій на кожному кроці узгоджується із загальною методикою МАІ, враховуючи, звичайно, вербальний характер вхідних даних.

Один із способів використання **якісних** (вербальних) **оцінок** та пов'язаних з ними нечітких множин наводиться у табл.2.

Таблиця 2 – Фазіфікація вербальних оцінок експерта

Інтенсивність	Якісна оцінка	Позначення якісної оцінки	Нечітка множина з відповідною якісною оцінкою
1	Однаково важливо	*ОВ*	{(1,0 / 1)}
3	*Ненабагато важливіше*	НВ	{(0,5 /1), (0,75 /2), (1,0 /3),(0,75 /4), (0,5 /5)}
5	Суттєво важливіше	СВ	{(0,5 /3), (0,75 /4), (1,0 /5),(0,75 /6), (0,5 /7)}
7	Значно важливіше	ЗВ	{(0,5 /5), (0,75 /6), (1,0 /7), (0,75 /8), (0,5 /9)}
10	Абсолютно важливіше	АВ	{(0,5 /9), (1,0 /10)}

Отже, маємо такі основні кроки:

Крок 1. Формування багаторівневої ієрархічної структури, яка містить інтегрований критерій, часткові критерії та об'єкти (проекти, стратегії) досліджування та впорядкування (вибору).

Крок 2. Побудова матриць попарних порівнянь з нечіткими оцінками для елементів, які знаходяться на окремих рівнях ієрархії.

Крок 3. Обчислення значень вагових коефіцієнтів (векторів пріоритетів) , кожного з елементів ієрархічної структури з погляду елемента, який перебуває на безпосередньо вищому рівні ієрархії.

Крок 4. Обчислення вектора пріоритетів , який визначає нечіткі оцінки аналізованих об'єктів (проектів, стратегій) з погляду інтегрованого критерію.

Крок 5. Впорядкування досліджуваних об'єктів (проектів) відносно величини нечітких оцінок гопт.

Формування багаторівневої ієрархічної структури критеріїв

Опишемо сутність операцій, здійснюваних на окремих кроках запропонованого Вітлінським В.В. алгоритму.

Загальний вигляд ієрархічної багатокритеріальної структури зображено на рис.1.

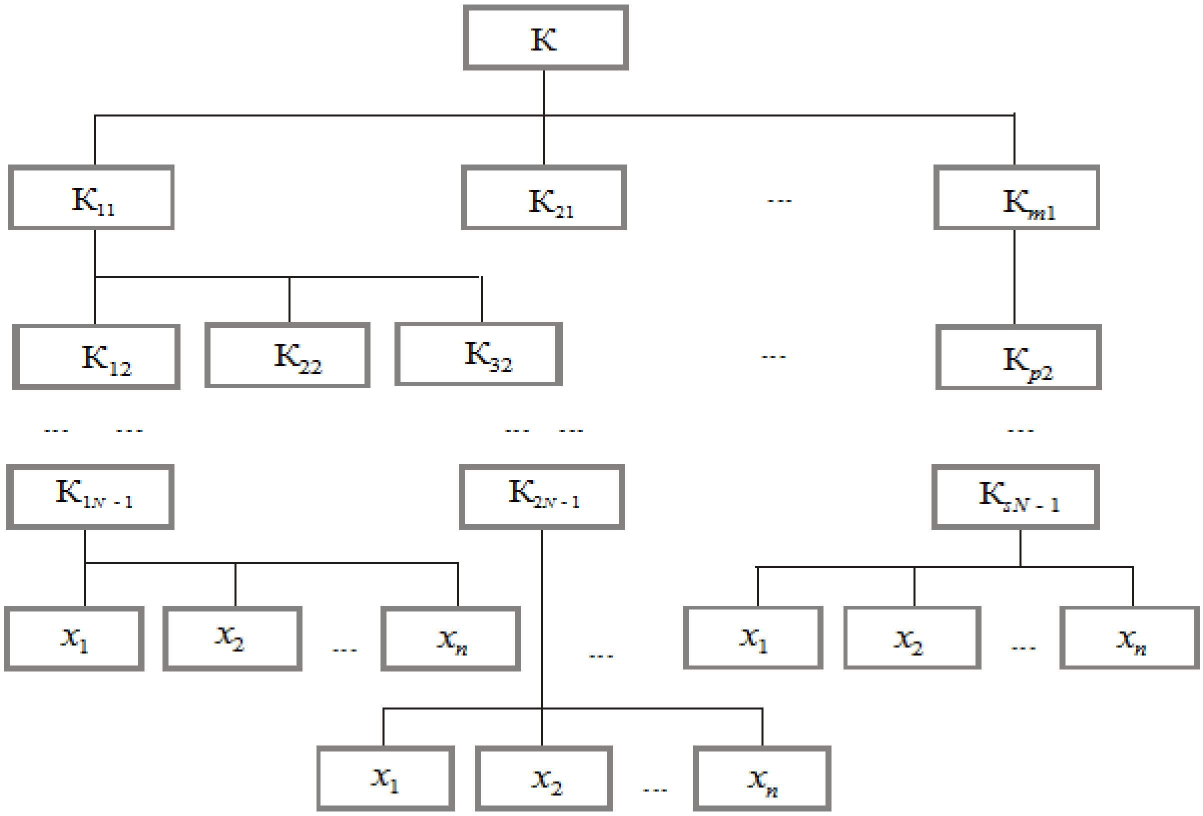


Рис.16.1 – Загальний вигляд багаторівневої ієрархічної структури

Позначення:

К — інтегрований критерій оцінювання; $К_{ij}$ — i-й критерій j-го рівня;

$i = 1, ..., m$ на рівні 1; $i = 1, ..., p$ на рівні 2; $i = 1, ..., s$ на рівні $N - 1$;

x_i — i-й об'єкт (проект), що аналізується, $i = 1, ..., n$;

m, p, s, n — кількість елементів відповідно на рівнях 1, 2, $N - 1$, N.

На верхньому рівні цієї структури (рівень 0) знаходиться лише один елемент — інтегрований критерій оцінювання, який можна розкласти (деталізувати) на кілька елементів (часткових критеріїв), тобто рівень 1, що йде безпосередньо за даним рівнем ієрархії. Кожний елемент цього рівня ієрархії, в свою чергу, деталізується на кілька елементів наступного рівня і т. д. На найнижчому рівні ієрархічної структури перебувають стратегії (об’єкти, проекти), які необхідно аналізувати та впорядковувати чи обирати один з них (елементи досліджуваної множини).

Побудована в такий спосіб ієрархічна багаторівнева структура дозволяє обмежитися відносно невеликою кількістю елементів на кожному рівні ієрархії та подолати проблеми, спричинені складністю інтегрованого критерію, що розглядається в багатьох випадках як критерій згортки.

Побудова матриці попарних порівнянь елементів з нечіткими оцінками

Для аналізу критеріїв оцінювання, що містяться на певних рівнях ієрархічної структури, пропонується побудова матриці попарних порівнянь елементів у вигляді:

$$\tilde{A} = \begin{pmatrix} & \tilde{A}_1 & \tilde{A}_2 & \dots & \tilde{A}_n \\ \tilde{A}_1 & \tilde{r}_1/\tilde{r}_1 & \tilde{r}_1/\tilde{r}_2 & \dots & \tilde{r}_1/\tilde{r}_n \\ \tilde{A}_2 & \tilde{r}_2/\tilde{r}_1 & \tilde{r}_2/\tilde{r}_2 & \dots & \tilde{r}_2/\tilde{r}_n \\ \dots & \dots & \dots & \dots & \dots \\ \tilde{A}_n & \tilde{r}_n/\tilde{r}_1 & \tilde{r}_n/\tilde{r}_2 & \dots & \tilde{r}_n/\tilde{r}_n \end{pmatrix},$$

де $\tilde{A}_1, \tilde{A}_2, \dots, \tilde{A}_n$ - нечіткі порівнювані елементи; $\tilde{r}_1, \tilde{r}_2, \dots, \tilde{r}_n$ - нечіткі вагові коефіцієнти (пріоритети) порівнюваних елементів; n – кількість порівнюваних елементів.

Побудовані у такий спосіб матриці порівнянь дають можливість здійснити попарне порівняння елементів на певному рівні ієрархічної структури з погляду їх важливості щодо критерію, який знаходиться на безпосередньо вищому рівні ієрархії і який є, власне, їх агрегованим критерієм (згорткою). При аналізі числових критеріїв (заданих на відповідних числових шкалах) можна обчислити окремі вагові коефіцієнти r та їх взаємне попарне відношення (ri/rj) у вигляді числових величин.

Враховуючи нестачу (відсутність) кількісних даних щодо оцінки відношення r*i*/r*j* (*i*, *j* = 1,...,*n*) пропонується відійти від прийнятого в MAI детермінованого підходу введенням та застосуванням лінгвістичного підходу, який ґрунтується на теорії нечітких множин. Для того, щоб одержати наведену вище нечітку матрицю попарних порівнянь, проводять опитування експерта відносно того, наскільки, на його думку, x*i* є вагомішим (значущим) для поняття, яке описується нечіткою множиною A, ніж елемент x*j*.

Таблиця 1 – Фазіфікація вербальних оцінок експерта

Інтенсивність	Якісна оцінка	Позначення якісної оцінки	Нечітка множина з відповідною якісною оцінкою
1	Однаково важливо	*ОВ*	{(1,0 / 1)}
3	*Ненабагато важливіше*	НВ	{(0,5 /1), (0,75 /2), (1,0 /3),(0,75 /4), (0,5 /5)}
5	Суттєво важливіше	СВ	{(0,5 /3), (0,75 /4), (1,0 /5),(0,75 /6), (0,5 /7)}
7	Значно важливіше	ЗВ	{(0,5 /5), (0,75 /6), (1,0 /7), (0,75 /8), (0,5 /9)}
10	Абсолютно важливіше	АВ	{(0,5 /9), (1,0 /10)}

У табл.1 наводяться поняття (лінгвістичні змінні), якими оперує експерт, інтерпретація цих понять — нечіткі множини $\tilde{m}_{ij} = \tilde{r}_i / \tilde{r}_j$, що можна представити одним з можливих переходів від вербального опису до нечіткого (фазіфікації). Визначення нечітких множин (фазіфікація), які репрезентують використовувані значення лінгвістичної змінної, досягається експертним шляхом. Після визначення нечітких множин, які репрезентують використовувані значення лінгвістичної змінної, можна здійснювати необхідні логічні та алгебраїчні операції з нечіткими множинами.

Отже, матриця попарних порівнянь (використовувана в МАІ) може бути модифікованою. Замість числових попарних відношень r*i*/r*j* вводяться вербальні (розпливчасті) відношення із позначкою "~" (i, j = 1,...,n). Тут символом «~» (тильда) позначено нечіткі категорії, тобто такі, які визначаються за допомогою нечітких множин. Маючи вербальні оцінки, можна сконструювати низку розпливчастих матриць попарних порівнянь.

Для ієрархічної структури, представленої на рис.16.1, це: на першому рівні — одна матриця для порівняння часткових критеріїв $К_{11}$, $К_{21}$,..., $К_{m1}$; на другому рівні – m матриць для порівняння часткових критеріїв $К_{12}$, $К_{22}$,..., $К_{p2}$ з погляду кожного з m критеріїв першого рівня; на рівні N – s матриць для порівняння n об'єктів з погляду кожного з критеріїв безпосередньо вищого рівня, тобто рівня (N-1).

Обчислення векторів нечітких ваг елементів ієрархічної структури

Для наведеної на рис.1 **ієрархічної структури** знаходимо, наприклад, вагові коефіцієнти критеріїв $К_{11}$, $К_{21}$, ..., $К_{m1}$ з погляду інтегрованого критерію К або критеріїв $К_{12}$, $К_{22}$, ..., $К_{p2}$ з погляду, наприклад, критерію $К_{11}$ тощо. Маючи на меті знаходження цих *вагових коефіцієнтів* r*i* застосовується **техніка середньої геометричної для нечіткої множини**.

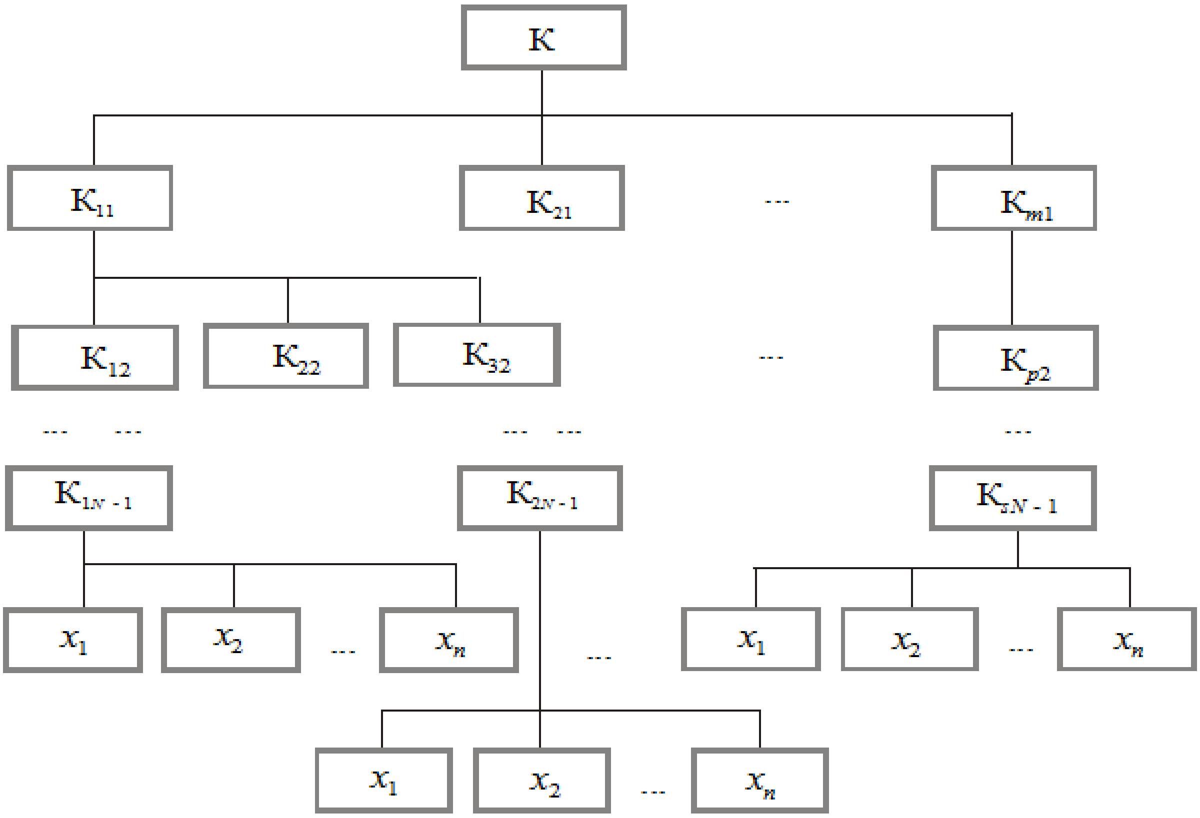


Рис.1 – Загальний вигляд багаторівневої ієрархічної структури

Знаходження вагових коефіцієнтів полягає в обчисленні середньої геометричної с*і* для елементів матриці А у такий спосіб:

$$\tilde{c}_i = \left(\prod_{j=i}^{n} \tilde{a}_{ij} \right)^{1/n} = \left(\tilde{a}_{i1} \ldots \tilde{a}_{in} \right)^{1/n}, \quad i = 1, \ldots, n, \qquad (1)$$

а також знаходженні нормалізованих нечітких вагових коефіцієнтів

$$\tilde{r}_i = \tilde{c}_i \Big/ \sum_{i=1}^{n} \tilde{c}_i , \quad i = 1, \ldots, n . \qquad (2)$$

Вони утворюють для кожної матриці порівнянь А певний **розпливчастий вектор пріоритетів** r=(r1,...,r*n*).

Обчислення (розпливчастого) вектора пріоритетів об'єктів (проектів) найнижчого рівня з погляду інтегрованого критерію (нульового рівня).

Вектор пріоритетів rk=(r1k,...,r*nk*) який визначає оцінки досліджуваних об'єктів x1,...,xn з погляду інтегрованого критерію К, згідно з методикою МАІ можна одержати множенням матриць, стовпчиками яких є вектори пріоритетів ряду поруч розташованих рівнів ієрархічної структури, відповідно з їх зв'язками, вказаними на рис.1.

Нехай розглядається деякий l-й рівень ієрархічної структури, елементи якого перебувають на безпосередньо вищому рівні щодо елементів рівня $l+1$, і одночасно вони знаходяться на один рівень нижче, ніж елементи рівня $l-1$.

Нехай маємо:

Y={y1,y2,...ym$l$} на рівні l; X={x1,x2,...xm$l+1$} на рівні $l+1$;

yj — j-й елемент l-го рівня; xi — i-й елемент $l+1$-го рівня;

z – елемент рівня l-1, якому безпосередньо підпорядковані всі елементи множини Y.

Нехай на рівні l-1 маємо певну функцію пріоритетів для відповідних елементів з рівня l:

$$\widetilde{r}_z : Y \to \{\widetilde{r}_z(y_i)\};$$

на рівні l маємо функцію пріоритетів для елементів l+1 рівня, які підпорядковані окремим елементам l-го рівня:

$$\widetilde{r}_{y_j} : X \to \{\widetilde{r}_{y_j}(x_i)\},\ \ j = 1,\ldots,m_l.$$

Отже, з погляду відповідних елементів рівня l-1 розпливчасті вагові оцінки рівня l+1 можна подати за такою формулою:

$$\widetilde{r}(x_i) = \sum_{j=1}^{m_l} \widetilde{r}_{y_j}(x_i)\cdot \widetilde{r}_z(y_j), i = 1,\ldots,m_{l+1}.$$

Якщо позначити через М$l+1$ матрицю з елементами

$$\{\widetilde{m}_{ij} = \widetilde{r}_{y_j}(x_i)\}$$

то маємо для трирівневої ієрархічної

$\widetilde{r}_x = \widetilde{M}_{l+1}\widetilde{r}$, де $\widetilde{r}_x$ *— вектор-стовпчик пріоритетів,*

що складається з елементів

$\widetilde{r}(x_i)$, а $\widetilde{r}$ — *вектор-стовпчик пріоритетів,*

що складається з елементів $\widetilde{r}_z(y_j)$.

Користуючись **методом математичної індукції**, одержимо для *N*-рівневої ієрархічної структури, зображеної на рис.1, вектор пріоритетів елементів найнижчого (*N*-го) рівня з погляду елемента (інтегрованого критерію К) найвищого рівня ієрархічної структури (0) у вигляді:

$$\widetilde{r}_K = \widetilde{M}_N \widetilde{M}_{N-1} \ldots \widetilde{M}_2 \widetilde{r},$$

де вектор-стовпчик

$$\widetilde{r}_\text{К} = (\widetilde{r}_{1\text{К}}, \widetilde{r}_{2\text{К}}, \ldots, \widetilde{r}_{n\text{К}})$$

репрезентує розпливчасті вагові оцінки аналізованих об'єктів з погляду інтегрованого критерію (К); r — вектор-стовпчик розпливчастих вагових коефіцієнтів елементів рівня 1 з погляду інтегрованого критерію (К);

Ml, l=2,..., N — матриці, вектор-стовпчиками яких є розпливчасті вектори пріоритетів відповідних елементів певного *l*-го рівня з погляду елемента безпосередньо вищого рівня (*l* – 1) ієрархічної структури (з урахуванням їх зв'язків, зазначених на рис.1).

Впорядкування досліджуваних об'єктів щодо величини нечітких оцінок

іcля реалізації третього і четвертого кроків алгоритму маємо нечіткі оцінки $\widetilde{r}_{i\text{К}}$ *i*=1,..., *n*, що являють собою розпливчасті вагові коефіцієнти аналізованих об'єктів з погляду інтегрованого критерію К.

На даному кроці необхідно порівняти між собою $\widetilde{r}_{i\text{К}}$, *i*=1,..., *n* для впорядкування об'єктів x1,...,xn відповідно до величин цих оцінок. Але, оскільки одержані оцінки є лише нечіткими множинами, то впорядкування об'єктів не є очевидним.

Таке впорядкування не можна виконати коректно, якщо спиратися лише на максимальні величини носіїв нечітких множин $\widetilde{r}_{i\text{К}}$ або лише на ті величини носіїв, яким відповідають максимальні ступені належності. Впорядкування лише відносно максимальних значень носіїв **нечіткої множини** не завжди призводить до коректного результату, бо великі значення носіїв можуть виступати з малими ступенями належності, і навпаки. Впорядкування лише на підставі тих значень носіїв, які мають максимальний ступінь належності, теж не завжди дає добрий результат, бо при цьому не враховуються всі інші елементи носія без урахування їх величин і відповідних їм значень функції належності. Умовою правильного впорядкування об'єктів є врахування як величин носіїв, так і їх ступеня належності у нечітких множинах $\tilde{r}_{iK}$, $i = 1, ..., n$.

Для цього в модифікованому алгоритмі доцільно використати, зокрема, **концепцію максимізуючої множини за Йєном**.

Максимізуючою множиною є така нечітка множина:

,

$$\tilde{\pi}(T) = \{(\mu_{\tilde{\pi}}(t))/t\}, \quad t \in T \quad (1)$$

де

$$\mu_{\tilde{\pi}}(t) = t / t_{\max}, \; t_{\max} = \sup T \quad (2)$$

T — множина всіх носіїв, які представляють розпливчасті оцінки аналізованих об'єктів x1,x2,...,xn.

Ступінь належності в максимізуючій множині $\mu_{\tilde{\pi}}(t)$ визначає ступінь близькості кожної величини носія до максимальної величини носія у множині *T*, тобто в множині всіх носіїв нечітких множин $\tilde{r}_{iK}$ які представляють оцінки об'єктів, що аналізуються з погляду інтегрованого критерію.

У межах кроку 5 виконуються такі етапи:

1) утворення максимізуючої множини;

2) формування для кожного об'єкта x1,x2,...,xn розпливчастої множини

$\widetilde{r}'_{i\mathrm{K}}$, яка є модифікованою оцінкою $\widetilde{r}_{i\mathrm{K}}$, тобто

$$\widetilde{r}'_{i\mathrm{K}} = \left\{\left(\mu_{\widetilde{r}'_{i\mathrm{K}}}(t)/t\right)\right\}, \quad (3)$$

де

$$\mu_{\widetilde{r}'_{i\mathrm{K}}} = \mu_{\widetilde{r}_{i\mathrm{K}}}(t)\Lambda\mu_{\widetilde{r}_{i}}(t),\ i = 1,\ldots,n, \quad (4)$$

L — оператор мінімума, що відповідає логічній операції «ТА»;

3) формування нечіткої множини $\widetilde{r}_{\text{опт}}$ — такої, що

$$\widetilde{r}_{\text{опт}} = \left\{\left(\mu_{\widetilde{r}_{\text{опт}}}(x_i)/x_i\right)\right\}, \quad (5)$$

де

$$\mu_{\widetilde{r}_{\text{опт}}}(x_i) = \vee_{i}\, \mu_{\widetilde{r}'_{i\mathrm{K}}},\ i = 1,\ldots,n, \quad (6)$$

V — оператор максимуму (логічне «АБО»), застосування якого приводить до того, що кожен об’єкт порівнюватиметься з іншими на підставі максимального ступеня належності в множині $\widetilde{r}'_{i\mathrm{K}}$, i=1,...,n.

У рамках даного етапу залежно від прийнятої системи гіпотез можливі різні модифікації. Зокрема, враховуючи ентропію

$$\left(H(x_i), i = 1,\ldots,n\right)$$

як міру невизначеності, формування нечіткої множини r*опт* можна виконати, якщо замість (6) скористатися таким виразом:

$$\mu_{\widetilde{r}_{\text{опт}}}(x_i) = \vee_{i}\, \mu_{\widetilde{r}'_{i\mathrm{K}}} /(1 + H(x_i)), i = 1,\ldots,n.$$

Для цього кожну з нечітких множин $\widetilde{r}'_{i\mathrm{K}}$ розподілимо на D_i, $i = 1,\ldots,n$ груп. Подібне групування слушно здійснити, зокрема, за принципом близькості у кожній нечіткій підмножині $\widetilde{S}_{d_i}$, $d_i = 1,\ldots,D_i$, $i = 1,\ldots,n$

відповідних значень носіїв.

Тобто, якщо $|t_j - t_k| \le \varepsilon$, то $\tilde{r}'_{i_j\text{К}}, \tilde{r}'_{i_k\text{К}} \in \tilde{S}_{d_i}$, $d_i = 1,\dots,D_i$, $i = 1,\dots,n$. Тут ε — задане число.

Приймаючи для кожної множини $\tilde{S}_{d_i}, d_i = 1,\dots,D_i, i = 1,\dots,n$

$$\mu_{d_i} = \vee_i \mu_{\tilde{S}_{d_i}},$$

одержимо

$$H(x_i) = -\sum_{d_i=1}^{D_i} \bar{\mu}_{d_i} \ln \bar{\mu}_{d_i}, \ i = 1,\dots,n,$$

де $\bar{\mu}_{d_i} = \mu_{d_i} \Big/ \sum_{d_i=1}^{D_i} \mu_{d_i}$, $d_i = 1,\dots,D_i$, $i = 1,\dots,n$;

4) **формування нечіткої множини** $\tilde{r}'_{\text{опт}}$ через відносну нормалізацію елементів нечіткої множини $\tilde{r}_{\text{опт}}$:

$$\tilde{r}'_{\text{опт}} = \{(\mu_{\tilde{r}'_{\text{опт}}}(x_i)/x_i)\},$$

де

$$\mu_{\tilde{r}'_{опт}}(x_i) = \mu_{\tilde{r}_{опт}}(x_i) / \max_i \mu_{\tilde{r}_{опт}}(x_i), i = 1,\dots,n$$

5) **впорядкування досліджуваних об'єктів** (проектів) x_i, $x_i, i = 1,\dots,n$ за величиною ступеня належності у множині $\tilde{r}_{\text{опт}}$ (від більшого значення серед $\mu_{\tilde{r}'_{\text{опт}}}(x_i)$, $i = 1,\dots,n$ до меншого).

Зауважимо, що отримані таким чином результати обчислень справедливі лише в рамках досліджуваної групи об'єктів (проектів, стратегій). За величиною ступеня належності в нечіткій множині $\tilde{r}'_{\text{опт}}$ можна обрати серед досліджуваних об'єктів той, для якого $\mu_{\tilde{r}'_{\text{опт}}} = 1$, а решту розташувати відповідно до спадання величини функції належності

$\mu_{\tilde{\gamma}'_{\text{опт}}}(x_i),\ i=1,\dots,n$.

Перевагою наведеного підходу є те, що за його допомогою відносно неважко проаналізувати причини отримання тих чи інших оцінок, використовуючи сформовану на кроці 1 ієрархічну структуру, аналізуючи (змінюючи), у разі необхідності, відповідні матриці якісних («м'яких») попарних порівнянь, верифікуючи вихідні судження.

Наведений алгоритм, який названо «**розпливчастим методом аналізу ієрархій**» (РМАІ), — ефективний при розв'язуванні проблем прийняття рішень з урахуванням ризику, які вимагають багатовимірних (багатокритеріальних) порівнянь, та коли складно чи неможливо одержати необхідні кількісні дані або процес здобуття кількісних даних потребує багато часу і зусиль, коштів, а натомість є можливість відносно просто дістати вербальні (описові) дані.

Не останньою перевагою **РМАІ** є можливість представлення вербальних даних у вигляді, зручному для комп'ютерної обробки інформації в системах підтримки прийняття рішень, обтяжених ризиком. Алгоритм зручний для створення інтерактивної інформаційної системи багатокритеріального аналізу за відсутності кількісної інформації, що дозволяє залучити кінцевого користувача (суб'єкта прийняття рішень) безпосередньо до процесу оцінювання варіантів, верифікації вербальних («м'яких») оцінок тощо.

РМАІ допускає ряд модифікацій залежно від прийнятої раціональної системи гіпотез, різних, адекватних ситуації, правил переходу від лінгвістичного опису альтернатив до розпливчастого (фазіфікації) тощо.

ЛІТЕРАТУРА

1. Леоненков, А. В. Нечеткое моделирование в среде MATLAB и fuzzyTECH / А.В. Леоненков. – СПб.: БХВ-Петербург, 2005. – 736 с.

2. Круглов, В. В. Нечеткая логика и искусственные нейронные сети / В. В. Круглов, М. И. Дли, Р. Ю. Голунов. – М.: ФИЗМАТЛИТ, 2001. – 201с.

3. Mamdani, E. H., Application of fuzzy algorithms for the control of a simple dynamic plant. In Proc IEEE (1974), 121–159.

4. Бабий, М. С. Применение элементов нечеткой логики для рейтинговой системы оценки знаний [Текст] / М. С. Бабий, А. П. Чекалов // Вісник СумДУ. Серія "Технічні науки". – 2011. – № 3. – С. 116-121.

5. Білоусова, Л. І. Методика обробки та інтерпретації результатів педагогічної діагностики [Текст] / Л. І. Білоусова, О. Г. Колгатін // Комп'ютер у школі та сім'ї. – 2003. – № 8. – С. 28-31.

6. Дуплик, С. В. Модели педагогического тестирования [Текст] [Электронный ресурс] / С. В. Дуплик. – Режим доступа : http://www.dupliksv. hut.ru /pauk/papers /testmodel.html.

7. Заде, Л. А. Понятие лингвистической переменной и его применение к принятию приближенных решений [Текст] / Л. А. Заде. – М. : Мир, 1976. – 167 с.

8. Иванова, О. Н. Выбор оптимальных методов обучения информатике учащихся средней школы на основе информационно-коммуникационных технологий. : автореф. дис. на соискание науч. степени канд. пед. наук : спец. 13.00.02 "Теория и методика обучения и воспитания (информатика)" [Рукопись] / О. Н. Иванова. – Челябинск., 2009. – 23 с.

9. Локтионова, Н. Н. Дифференцированное обучение математике с использованием контроля качества подготовки студентов, основанное на элементах нечеткой логики : автореф. дис. на соискание науч. степени канд. пед. наук : спец. 13.00.02 "Теория и методика обучения и воспитания (математика)" [Рукопись] / Н. Н. Локтионова. – М., 2011. – 22 с.

10. Монахов, В.М. О возможностях методологии нечеткого моделирования как нового инструментария информатизации педагогических объектов [Текст] / В. М. Монахов // Современные информационные технологии и ИТ-образование : сб. избр. трудов Междунар. науч.-практ. конф. – М. : МГУ, 2008. – С. 28-32.

11. Федієнко, В.В. Моделі кваліметрії і порівняння рівнів навчальних досягнень студентів у різних оціночних системах : автореф. дис. на здобуття наук. ступеня канд. пед. наук : спец. 13.00.09 "Теорія навчання" [Рукопис] / В. В. Федієнко. – Харків, 2009. – 23 с.

12. Цідило, І.М. Нечіткість та невизначеність : опис, вимірювання і способи вирішення в моделюванні педагогічних явищ [Текст] [Електронний ресурс]. – Режим доступу : http://archive.nbuv.gov.ua/e-journals/ITZN/2012_5/708-2338-1-ED.pdf.

13. Штовба, С.Д. Введение в теорию нечетких множеств и нечеткую логику [Текст] [Электронный ресурс]. – Режим доступа : http://matlab.exponenta. ru/fuzzylogic/book1/.

14. Zadeh, L. A. A fuzzy-set-theoretic interpretation of linguistic hetges / L. A. Zadeh. J. Gybern, 1972. – № 2. – P. 4-34.

15. Drechsel, D. Regelbasierte Interpolation und Fuzzy Control. - Braubschweig/Wiesbaden: Friedr. Vieweg&Sohn Verlagsgesellschaft mbH. - 1996.

16. Passino K. M., Yurkovitch. Fuzzy Control. - ADDISON-WESLEY. - 1998.

17. Vogel J. u.a. Elektrische Antriebstechnik. – Berlin: Verlag Technik GmbH. - 1991.

18. Kahlert J., Frank H. Fuzzy-Logik und Fuzzy-Control. Eine anwendungsorientierte Einführung mit Begleitsoftware. - Braubschweig/Wiesbaden: Friedr. Vieweg&Sohn Verlagsgesellschaft mbH. - 1993.

19. Калашников В.И., Палис Ф. Введение в интелектуальные системы программного управления – Донецк: ДонГТУ. - 1997.

20. Fuzzy Logic Toolbox User’s Guide: MathWorks. - 1998.

21. Brause R. Neuronale Netze: eine Einführung in die Neuroinformatik. - Stuttgart: Teubner. - 1991.

22. Kruse R., Gebhardt J., Klawonn F. Fuzzy-Systeme.- Stuttgart: Teubner. - 1993.

23. Nauk D., Klawonn F., Kruse R. Neuronale Netze und Fuzzy-Systeme, Grundlagen des Konnektionismus, Neuronaler Fuzzy-Systeme und der Kombination mit wissensbasierten Methoden. - Braunschweig: Teubner. - 1992.

24. Калашников В.И., Денисенко И.В. Теория нейросетей. - Донецк: ДонГТУ. - 1997.

25. Матвійчук А. В. Штучний інтелект в економіці : нейронні мережі, нечітка логіка: [монографія] / А. В. Матвійчук. – Київ: КНЕУ, 2011. – 439 с.

26. Гаврилова Т. А. Базы знаний интеллектуальных систем / Т. А. Гаврилова, В. Ф. Хорошевский. – СПб.: Питер, 2001. –384 с.

27. Бакан Г. М. Вступ до теорії експертних систем та баз знань / Г. М. Бакан. – К.: ВПЦ «Київський університет», 2005. – 90 с.

28. Дорф Р. Современные системы управления / Р. Дорф, Р. Бишоп; [пер. с англ.]. – М.: Лаборатория Базовых Знаний, 2002 – 832 с.

29. Каллан Р. Основные концепции нейронных сетей / Р. Каллан. – М.: Вильямс, 2001. – 288 с.

30. Ануфриев И. Е. Информатика. Пакет MatLab / И. Е. Ануфриев. – Изд-во СПбГПУ, 2003. – 67 с.

31. Иглин С. П. Математические расчеты на базе MATLAB / С. П. Иглин. – СПб.: БХВ-Петербург, 2005. – 640 с.

32. Кудрявцев Д. Многогранность управления знаниями на предприятии [Электронный ресурс] / Д. Кудрявцев. – Режим доступа: // ttp://www.big.spb.ru/publications/bigspb/km/mnogogr_uz_na_predpr.shtml

33. Бессмертный И. А. Искусственный интеллект: [учебное пособие] / И. А. Бессмертный. – СПб: СПбГУ ИТМО, 2010. – 132 с.

34. Нечеткая логика: алгебраические основы и приложения: [монография] / С. Л. Блюмин, И. А. Шуйкова, П. В. Сараев, И. В. Черпаков. – Липецк: ЛЭГИ, 2002. – 113 с.

35. Борисов А. Н. Принятие решений на основе нечетких моделей. Примеры моделей / А. Н. Борисов, О. А. Крумберг, И. П. Федоров. – Рига: Зинатне, 1990. – 184 с.

36. Ротштейн А. П. Интеллектуальные технологи идентификации: нечеткая логика, генетические алгоритмы, нейронные сети / А. П. Ротштейн. – Винница: УНІВЕРСУМ – Вінниця, 1999. – 320 с.

37. Рутковская Д. Нейронные сети, генетические алгоритмы и нечеткие системы / Д. Рутковская, М. Пилинский, Л. Рутковский. – М.: Горячая линия – Телеком, 2004. – 452 с.