

УДК 62:50

М. Корчемний, докт. техн. наук; В. Федорейко, докт. техн. наук

*Тернопільський національний педагогічний університет
імені Володимира Гнатюка*

ОСНОВИ МЕТОДИКИ РОЗРОБКИ ІНТЕЛЕКТУАЛЬНИХ СИСТЕМ УПРАВЛІННЯ ТА ПРИЙНЯТТЯ РІШЕНЬ НА ОСНОВІ НЕЧІТКОЇ ЛОГІКИ В СЕРЕДОВИЩІ MatLAB

Викладено основи методики розробки інтелектуальних систем управління і прийняття рішень на основі нечітких множин і логіки. Ефективність ідентифікації об'єктів нечіткими системами при різних способах їх навчання ілюстровано на прикладі створення експертної системи для оцінки екологічного потенціалу регіону. Показано, що нечіткі системи Сугено, налаштовані за допомогою алгоритмів навчання нейронних мереж, мають найвищу точність. При великому числі змінних параметрів (налаштування) цільової функції, для її оптимізації доцільно використовувати генетичні алгоритми.

N. Korchemny, V. Fedoreyko

BASES OF A TECHNIQUE OF DEVELOPMENT OF INTELLECTUAL CONTROL SYSTEMS AND ACCEPTANCE OF THE DECISIONS ON THE BASIS OF INDISTINCT LOGIC IN ENVIRONMENT MatLAB

The bases of a technique of development of intellectual control systems and acceptance of the decisions are stated on the basis of indistinct sets and logic. The efficiency of identification of objects by indistinct systems at different ways of their training is illustrated on an example of creation of expert system for an estimation of potential of ecological tourism of region. It is shown, that the indistinct systems Sugeno, adjusted with the help of algorithms of training neural of networks, have the best accuracy. At the large number variable (parameters of adjustment) it is expedient to use criterion functions, for its optimization genetic algorithms.

Вступ. Керування сучасним виробництвом і окремими технологіями в умовах інтенсивної зміни навколишнього середовища, невизначеності і недостатньо прогнозованого майбутнього є складним, динамічним процесом. Різноманітність цілей і задач, що виникають в процесі прийняття рішень, їх складність і часові обмеження, характерні для багатьох проблем, для яких приймаються рішення, вимагають комп'ютерної підтримки цього процесу. Тому створення таких систем підтримки рішень, які забезпечують менеджера (оператора) сучасними засобами аналізу інформації, генерації варіантів рішень, їх оцінками і вибором найкращого варіанту, є важливою і актуальною задачею.

Традиційні методи управління передбачають достатньо точний аналітичний опис динаміки процесу і середовища. Тому такими методами часто складно, а інколи і неможливо забезпечити необхідні результати управління нелінійними об'єктами, функціонуючими в середовищі з різними збуреннями і флуктуаціями. Інколи навіть при наявності придатної аналітичної моделі початкові параметри об'єкта можуть бути неповними, що не забезпечує адекватність управління його процесами.

У цьому плані прикметною є точка зору Лотфі Заде: "Я вважаю, що надмірне прагнення до точності стало справляти дію, що зводить нанівець теорію управління і теорію систем, оскільки воно приводить до того, що дослідження в цій області зосереджуються на тих і тільки тих проблемах, які підлягають точному розв'язку. В результаті багато класів важливих проблем, в яких дані цілі і обмеження, є складними або погано визначеними для того, щоб застосувати точний математичний аналіз, залишались і залишаються осторонь з тієї причини, що вони не піддаються математичній трактовці. Для того, щоб сказати дещо суттєве для проблем подібного роду, ми повинні відмовитись від наших вимог точності і допустити результати, які є

дещо розмитими або невизначеними" [1]. Ця робота Л. Заде започаткувала один із напрямів інтелектуальних технологій – нечітку логіку. До інтелектуальних технологій відносять алгоритми обробки даних, які відтворюють в тому або іншому відношенні логіку міркувань людини. Інтелектуальні технології усувають значну кількість труднощів, властивих класичним методам управління завдяки можливості синтезування систем з високою пристосованістю до існуючих і непередбачуваних змін, в динаміці об'єкта керування [2].

Зростаюча складність отримуваної інформації розв'язок багатьох проблем управління сучасним виробництвом на основі обробки неповної, неточної або суперечливої інформації, прийняття рішень в умовах, коли знання важко або неможливо формалізувати, а основні критерії і залежності погано або неоднозначно визначені, вимагають подальшого розвитку і адаптації інтелектуальних технологій до прикладних задач управління та прийняття рішень.

В інтелектуальних технологіях виділяються два підходи до моделювання складних нелінійних систем: лінгвістичний і чисельний. Лінгвістичний підхід засновується на нечіткій логіці, основою якої є теорія нечітких множин [1, 3], а чисельний – на штучних нейронних мережах [3, 4], здатних до самонавчання, і на генетичних алгоритмах [4, 5], синтезуючих рішення керування шляхом еволюційних обчислень.

Нечітка логіка, на якій засновується нечітке керування, ближча за суттю до мислення людини і природних мов, ніж традиційні логічні системи. Вона, в основному, забезпечує ефективні засоби відображення невизначеностей і неточностей. Математичний апарат нечіткої логіки призначений для виявлення і формалізації досвіду людини при вирішенні складних проблем. В системах управління та прийняття рішень методи нечіткої логіки застосовуються при керуванні режимами складних технологічних процесів, в задачах діагностики роботи обладнання, в алгоритмах опосередкованого контролю невимірюваних змінних тощо. Вихідною інформацією при розробці алгоритму служать дані спеціально організованого опитування досвідчених операторів або спеціалістів, за результатами яких після спеціальної обробки формується алгоритм.

Окрім безперечних переваг, нечіткі системи мають також суттєві недоліки: вони не володіють здатністю навчання й адаптації їх до зміни ситуацій, а також вигляд і параметри функцій належності вибираються в них суб'єктивно і можуть виявитися не тими, які повністю відображають реальну дійсність. Тому для корекції параметрів функцій належності і адаптації нечітких систем в процесі їх роботи останнім часом застосовуються нейронні мережі і генетичні алгоритми.

Нейронні мережі, обчислювальні структури спеціальної конфігурації володіють властивостями навчання. На початковій стадії розробки прикладного алгоритму мережа навчається на пред'явлених на її входи зразках – масивах вихідних даних і відповідних реакцій. Вони застосовуються в задачах розпізнавання зорових образів (зокрема, в роботах), при реалізації функцій діагностики стану обладнання, ідентифікації об'єктів управління, непрямого контролю тощо.

Основною перевагою нейромережного підходу є можливість виявлення закономірностей в даних, їх узагальнення, тобто видобування знань з даних, а основним недоліком – неможливість безпосередньо (в явному вигляді, а не у вигляді вектора вагових коефіцієнтів міжнейронних зв'язків) представити функціональну залежність між входом і виходом досліджуваного об'єкта. Іншим недоліком нейромережного підходу є трудність формування презентабельної вибірки, велике число циклів навчання і забування "старого" досвіду, складність визначення розміру і структури нейронної мережі. Тому інтеграція цих двох незалежних інтелектуальних технологій: штучних нейронних мереж і теорії нечітких множин та логіки – дозволяє створити нову, більш універсальну методологію обробки різного виду знань і моделювання складних процесів.

При налаштуванні нейронних мереж і нечітких систем, коли за заданими входами отримують бажані виходи, шляхом підбору ваг міжнейронних зв'язків або параметрів функцій належності та ваг правил нечіткого висновку, отримують для оптимізації складну цільову функцію. Велика кількість синтезованих параметрів та неможливість аналітичного представлення залежності вибраної функції мети від цих параметрів унеможлиблює застосування градієнтних методів пошуку для розв'язання поставленої задачі. Такі задачі розв'язуються за допомогою біологічних еволюційних підходів до оптимізації - генетичних алгоритмів. При цьому кожний варіант (набір значень змінних) розглядають як особину, а значення цільової функції для цього варіанту – як міру пристосовуваності даної особини. Тоді в процесі еволюції пристосовуваність особин буде зростати. Генетичний алгоритм – це послідовність керованих дій і операцій, моделюючих еволюційні процеси на основі аналогів механізмів генетичного успадкування і природного відбору. При цьому зберігається біологічна термінологія: хромосома-вектор з нулів і одиниць, кожна позиція якого називається геном; набір особин – популяція; кросовер-операція, при якій дві хромосоми обмінюються своїми частинами; мутація – випадкова зміна однієї або декількох позицій в хромосомі. Генетичний алгоритм імітує еволюцію популяції як циклічний процес схрещування особин, мутації і зміни поколінь. Протягом життєвого циклу популяції, тобто в результаті випадкових схрещувань (за допомогою кросоверу) і мутацій, до неї додається деяка кількість нових варіантів. Далі проходить відбір, відбираються особини, які найбільше відповідають значенню цільової функції, і формується нова популяція. До нової популяції знову відбирають кращі особини і так далі. Кожне наступне покоління буде кращим від попереднього. Коли пристосовуваність особин перестане покращуватись, процес зупиняється і як розв'язок задачі оптимізації приймається найкращий із знайдених варіантів. Застосування генетичних алгоритмів для налаштування нечітких систем і нейронних мереж дозволяє імітувати роботу інженерних об'єктів будь-якої складності та розробляти ефективні інтелектуальні системи керування режимами їх роботи.

Розвиток інформаційних комп'ютерних технологій сприяв створенню значної кількості прикладних програм, які реалізують різні методи інтелектуальних технологій [7]. При розв'язанні складних задач управління з високою точністю при погано визначених умовах або станах нелінійних об'єктів керування опціонально достатнім є програмне середовище MatLAB зі спеціальними програмними пакетами Fuzzy Logic Toolbox, Neural Networks Toolbox і Genetic Algorithm and Direct Search Toolbox [8].

Постановка задачі. Аналіз літературних джерел показує відсутність єдиної методології розробки інтелектуальних систем управління, спроможних моделювати і аналізувати різні види знань (кількісні, нечіткі, неповні і неоднозначні) та здатних до навчання на основі накопиченого досвіду і адаптації до змін зовнішнього середовища та умов функціонування об'єкта керування. Перспективним напрямом в розробці інтелектуальних систем управління та прийняття рішень є інтеграція методів штучного інтелекту: нейронних мереж, нечітких множин і генетичних алгоритмів. Метою даних досліджень є розробка методики ідентифікації і моделювання складних нелінійних об'єктів та створення систем управління ними на основі інтеграції методів інтелектуальних технологій.

Методика проектування нечітких систем. Нечітка множина A , як частина універсальної множини X , – це набір пар $\{(x, \mu^A(x))\}$, де $x \in X$ і $\mu^A : X \rightarrow [0, 1]$ – функція належності (ФН), що являє собою деяку суб'єктивну міру відповідності елемента нечіткій множині;

$\mu^A(x)$ може набувати значення від нуля, що означає абсолютну неналежність елемента до нечіткої множини, до одиниці, що означає повну належність елемента x до нечіткої множини A .

Лінгвістична змінна – це змінна, значення якої є слова або словосполучення. На області свого визначення ця змінна задається терм-множинами або термами, кожна з

яких є нечіткою змінною. Нечітка змінна – це нечітка множина, визначена в частині області визначення лінгвістичної змінної і задана у вигляді ФН.

Для кращого розуміння короткого викладу теорії нечітких систем будемо ілюструвати їх застосування на прикладі. Нехай потрібно створити нечітку систему для експертної оцінки інвестиційної привабливості регіону як місця розвитку екотуризму (в даному прикладі оцінюється як об'єкт екотуризму с. Скоморохи та його околиці, що на Тернопільщині). В основному величина екотуристичної привабливості оцінюється за такими критеріями [9]:

- стан комунальної сфери (KS);
- стан сфери послуг (SP);
- розташування та транспортна доступність (TD);
- природні рекреаційні туристичні ресурси (PRTR);
- антропогенні туристичні ресурси (ATR);
- духовні та релігійні пам'ятки (DRP).

Кожний критерій для конкретного регіону оцінюється від 0 до 10 балів. Після оцінки значень критеріїв, за методикою [9] визначається рівень привабливості регіону в межах від 0 до 100 балів. Створення експертної системи дозволяє автоматизувати процедуру оцінки рівня привабливості регіону і зменшує витрати праці та часу.

Названі вище критерії та вихід-рівень привабливості приймаються як лінгвістичні змінні. Задамо кожну вхідну лінгвістичну змінну 5, а вихідну 7 терм-множинами гаусівського типу, розташованими рівномірно на універсумі визначення лінгвістичної змінної.

Терм-множини для вхідних лінгвістичних змінних: n – низький, ns – нижче середнього, s – середній, vs – вище середнього, v – високий. Для вихідної змінної додається ще два крайні терми: dn – дуже низький і dv – дуже високий.

Вигляд і послідовність обробки інформації системою нечіткого висновку показано на рисунку 1.

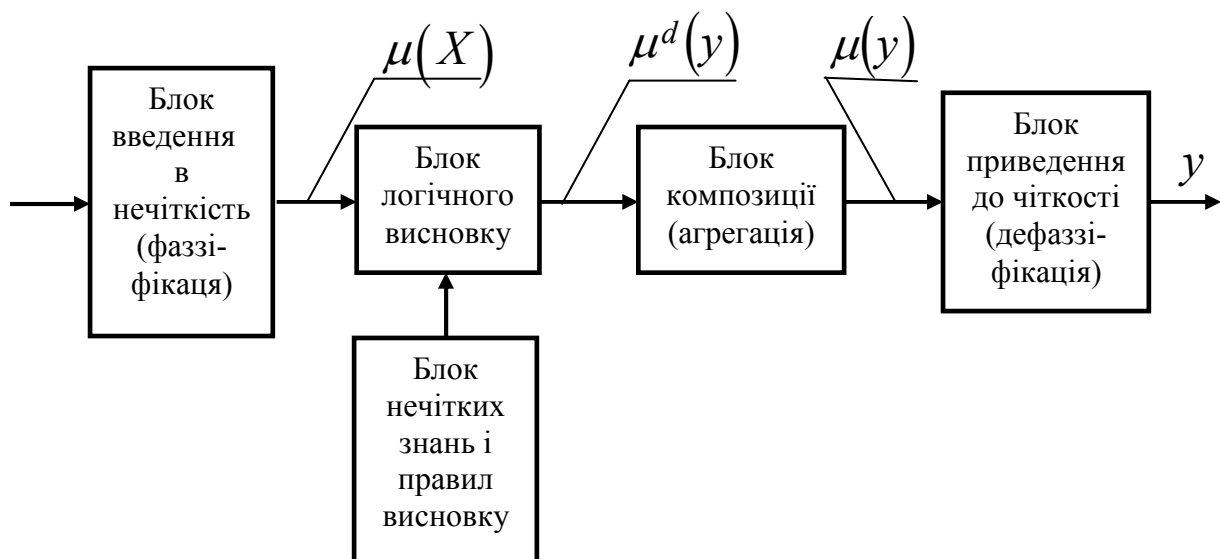


Рисунок 1 - Структура системи нечіткого висновку (нечіткої системи)

Блок введення в нечіткість оцінює ступінь належності значень вхідних лінгвістичних змінних, заданих кількісно або якісно, до окремих їх терм-множин.

Побудувати нечітку систему можна програмно або за допомогою редактора нечіткого висновку FIS [10, 12], який відкривається з командного вікна MatLAB функцією fuzzy. У пункті меню File редактора слід вибрати операцію: New Fis і встановити тип системи нечітково висновку – Mamdani або Сугено. При завантаженні раніше створеної системи вибирають операцію Import і імпортують нечітку систему з

робочої області (From Workspace...) або із зовнішнього файлу (From Disk...), активізуючи блоки цих змінних, задаються їх назви, діапазони визначення (вхідні – [0, 10]), вихідна – [0, 100], а команда Membership Functions – викликає редактор функцій належності з якого задаються вид і кількість терм-множин. Для прикладу на рисунку 2 наведено вигляд функцій належності для вхідних KS і SP та вихідної КО змінних.

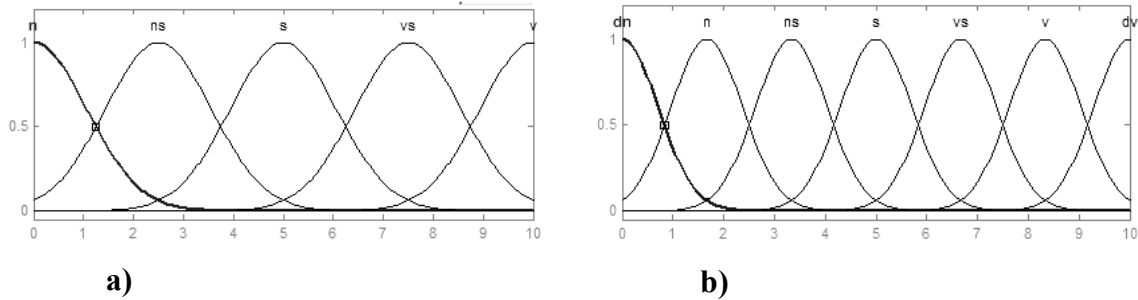


Рисунок 2 - Гаусівські функції належності вхідних KS і SP (а) та вихідної КО (в) змінних

База нечітких знань і правил (рисунок 1) формується за типом: ЯКЩО...ТОДІ, і являє собою сукупність поєднаних нечітких значень вхідних лінгвістичних змінних, яке відповідає конкретній терм-множині вихідної лінгвістичної змінної. Якщо ми маємо 6 лінгвістичних змінних і кожна задана 5 терм-множинами, то загальна кількість правил (комбінацій сполучень термів) складає 5^6 . Для подолання так званого "прокляття" розмірності для складних систем потрібно будувати ієрархічні системи нечіткого висновку (рисунок 3), при цьому кількість правил зменшується більше ніж у 50 разів.

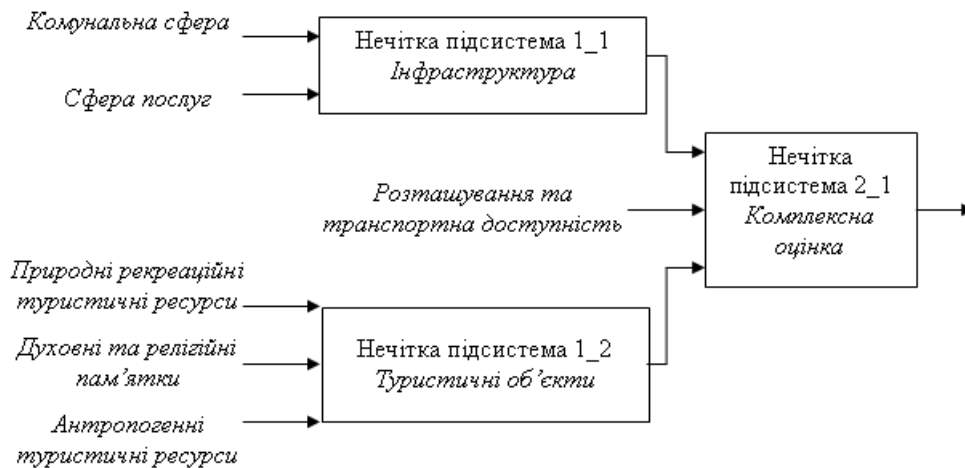


Рисунок 3 - Ієрархічна нечітка система оцінки екотуристичного потенціалу селища

Найбільш поширеними алгоритмами нечіткого висновку є алгоритми Мамдані і Сугено.

Правила для нечіткої підсистеми 1_1 за Мамдані:

$$\begin{aligned}
 \text{П1: ЯКЩО } KS \in n \text{ I } SP \in n, \text{ ТОДІ } \textit{infra} \in dn \text{ (1);} \\
 \text{П2: ЯКЩО } KS \in n \text{ I } SP \in ns, \text{ ТОДІ } \textit{infra} \in n \text{ (1);}
 \end{aligned} \tag{1}$$

⋮

$$\text{П25: ЯКЩО } KS \in v \text{ I } SP \in v, \text{ ТОДІ } \textit{infra} \in dv \text{ (1),}$$

де в дужках після кожного правила записана його вага.

Правила нечіткого висновку (1) складаються з двох частин: умови, яка записується між словами ЯКЩО і ТОДІ та, власне, висновку, який записується після слова ТОДІ. Істинність умови визначається для кожного правила (1) за виразом:

$$\mu_j(X) = w_j \min_{i=1,2}(\mu^j(x_i)), \quad j=1,25, \quad i=1,2, \tag{2}$$

де j - е правило; $i=1,2$ - кількість вхідних змінних; $x_i=(X_{ks}, X_{sp})$; $\mu^j(x_i)$ - функції належності умов j - правила вхідних змінних (KS, SP); X_{ks}, X_{sp} - конкретні оцінки критеріїв SP, KS .

Операція мінімум у виразі (2) виконується у випадку, коли підумови типу $KS \in n$ і $SP \in n$ з'єднуються логічним І, при з'єднанні підумов логічним АБО виконується операція максимум.

Результат нечіткого висновку за j -им правилом бази знань (1) визначається за виразом:

$$\mu^{d_j^*}(y) = \min(\mu^{d_j}(y), \mu_j(x)), \quad (3)$$

де y - ом позначено вихід підсистеми *infra*; $\mu^{d_j}(y)$ - функція належності терма вихідної лінгвістичної змінної у j -го правила.

Результат логічного висновку за всією базою знань (1) знаходиться агрегуванням нечітких множин (3)

$$\mu(y) = \max_{j=1,25}(\mu^{d_j^*}(y)), \quad (4)$$

де $\mu(y)$ - результуюча функція належності вихідної лінгвістичної змінної *infra* при заданому векторі входу $X=(x_{ks}, x_{sp})$. Чітке значення виходу *infra*, відповідне вектору входу X , визначається через дефаззифікацію нечіткої множини (4):

$$\text{inf ra} = y = \frac{\int_0^{100} y\mu(y)dy}{\int_0^{100} \mu(y)dy}. \quad (5)$$

Згідно з (2), (3), (4) і (5), у FIS-редакторі необхідно встановити: An method-min, Or method-max, Implication-min, Aggregation-max, Defuzzification-centroid.

При розробці нечіткої системи Сугено правила (1) видозмінюються в частині висновку. Наприклад, для j -правила висновок (після слова ТОДІ) записується у вигляді:

$$y_j = a_j x_{ks} + b_j x_{sp}, \quad (6)$$

де a_j, b_j - деякі вагові коефіцієнти.

За виразом (6) при відомих оцінках вхідних змінних KS і SP розраховуються виходи кожного правила. Істинність умов кожного правила визначається за виразом (2).

Чітке значення вихідної змінної підсистеми визначається за формулою (при 25 правилах):

$$y = \frac{\sum_{j=1}^{25} y_j \cdot \mu_j(X)}{\sum_{j=1}^{25} \mu_j(X)}. \quad (7)$$

Після складання правил (1) відкривається редактор правил у FIS-редакторі (Edit/Rules або View/Rules), записуємо їх у вікно редактора. Після запису правил і виконанні операції Surface в меню View у вікні редактора відобразиться залежність (поверхня) $\text{Infra} = f(KS, SP)$.

Порівняння отриманих нечіткою підсистемою оцінок інфраструктури (*outff*) з оцінками експертів (*out*) здійснюється за допомогою програми [12] і наведена на рисунку 4. Середня похибка оцінки інфраструктури нечіткою системою складає 0,398.

Створену систему нечіткого висновку за допомогою операції Export (пункт меню File) зберігають під іменем *Infra* в робочому просторі *To Workspace*.

Таким же чином створюють нечіткі підсистеми 1_2 - Туристичні об'єкти і 2_1 - Комплексна оцінка і зберігають в робочому середовищі *MatLAB* під назвами,

відповідно, Tur.Ob і КО. Середні похибки відтворення дійсних оцінок Tur.Ob і КО складають, відповідно, 0.45318 і 4.5318.

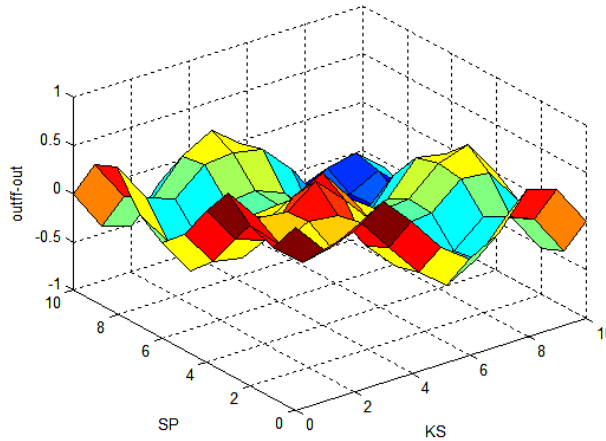


Рисунок 4 - Поверхня похибок оцінки інфраструктури нечіткою системою Infra в порівнянні з оцінками експертів при різних значеннях входів SP і KS

Для створення моделі експертної системи в Simulink у вікно створення моделі переносять три блоки Fuzzy Logic Controller з бібліотеки Fuzzy Logic Toolbox і з'єднують їх у відповідності ієрархії рисунка 2. У вікна налаштування блоків записують назви створених у FIS-редакторі нечітких підсистем 1_1, 1_2 і 2_1, до входів під'єднують блоки Const (рисунок 5).

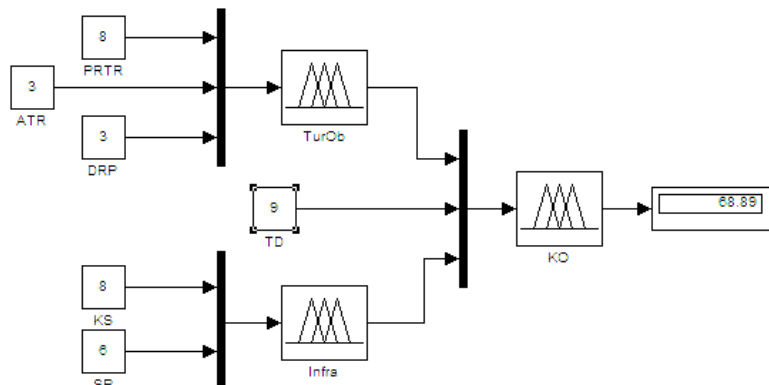


Рисунок 5 - Структурна схема моделі експертної системи оцінки привабливості регіону для екотуризму на основі нечіткої логіки

Задавши за оцінками експертів значення вхідних змінних (6 критеріїв) в блоках Const і запустивши модель в роботу, отримаємо на виході у блоці Display рівень привабливості регіону для екотуризму в межах [0, 100].

Комплексна оцінка екотуристичної привабливості села Скоморохи і його околиць, визначена експертною системою на основі алгоритму Мамдані, склала 44.44 ± 4.53 . За вищевикладеною методикою за допомогою редактора FIS створено нечітку експертну систему, підсистеми якої (рисунок 3) реалізовані на основі алгоритму Сугено. Оцінка екотуристичної привабливості регіону за цією системою склала 44.44 ± 1.0767 .

Навчання нечітких систем. Навчання (налаштування) являє собою ітераційну процедуру знаходження таких параметрів функції належності і ваги правил, які мінімізують відхилення результатів нечіткого висновку від експериментальних даних. При цьому проводиться пошук мінімуму середньоквадратичного відхилення для системи Мамдані [13]:

$$\sqrt{\frac{1}{M} \sum_{r=1, M} (y_r - F(P, W, X_r))^2} \rightarrow \min, \quad (8)$$

для системи Сугено:

$$\sqrt{\frac{1}{M} \sum_{1,M} (y_r - F(P, B, X_r))^2} \rightarrow \min, \quad (9)$$

де $X_r = (x_{r,1}, x_{r,2}, \dots, x_{r,n})$ – вхідний вектор в r -ій навчальній вибірці, y_r – відповідний вихід (Для підсистеми *Infra* $X_r = (x_{r,KS}, x_{r,SP})$); P – вектор параметрів функції належності термів вхідних і вихідних змінних (оскільки прийняті гаусові функції належності $\mu(x) = e^{-\frac{(x-b)^2}{2c^2}}$), то для *Infra*

$$P = [c_{n,KS}, c_{ns,KS}, \dots, c_{v,KS}, c_{n,SP}, c_{ns,SP}, \dots, c_{v,SP}, c_{dn,y}, c_{n,y}, \dots, c_{v,y}, c_{dv,y}, b_{n,KS}, b_{ns,KS}, \dots, b_{v,KS}, b_{n,SP}, b_{ns,SP}, \dots, b_{v,SP}, b_{dn,y}, b_{n,y}, \dots, b_{v,y}, b_{dv,y}],$$

де y -м позначено вихід підсистеми *infra*; W – вектор вагових коефіцієнтів правил бази знань. Для (1) $W_{infra} = (w_1, w_2, \dots, w_{24}, w_{25})$;

$F(P, W, X_r)$ – результат висновку за нечіткою базою Мамдані з параметрами (P, W) при значенні входів X_r ; B – вектор коефіцієнтів лінійних функцій у висновках правил бази знань Сугено;

$F(P, B, X_r)$ – результат висновку за нечіткою базою Сугено з параметрами (P, B) для вхідного вектора X_r .

Аналітично для підсистеми *Infra* результат висновку записується у вигляді:

$$\begin{aligned} \mu^j(x_i) &= e^{-\left(\frac{x_i - b_i^j}{c_i^j}\right)^2}; \\ \mu^{dj}(y) &= e^{-\left(\frac{y - b^k}{c_k}\right)^2}; \\ \mu(y) &= \max_{j=1,25} \left\{ \min_{j=1,25} \left[\mu^{dj}(y), \left(w_j \min_{i=1,2} (\mu^j(x_i)) \right) \right] \right\}; \\ \text{Infra} = y &= \frac{\int_0^{10} y \mu(y) dy}{\int_0^{10} \mu(y) dy}. \end{aligned} \quad (10)$$

Для нечіткого висновку Сугено нульового порядку:

$$\begin{aligned} \mu^{dj}(y) &= \mu^{dj}(X^*) = \min_{i=1,2} (\mu^j(x_i)); \\ \text{Infra} = y &= \frac{\sum_{j=1,25} \mu^{dj}(X^*) y_j}{\sum_{j=1,25} \mu^{dj}(X^*)}, \end{aligned} \quad (11)$$

де y_j – точки на осі визначення оцінки *Infra*, в яких задані терми у вигляді синглотонів (імпульсів одиничної висоти і нульової ширини).

Навчання системи нечіткого висновку Сугено здійснюється за допомогою програми *anfis* [11,13].

Навчання за допомогою програми *anfis* перетворює нечітку систему в так звану гібридну мережу, в якій висновки робляться на основі апарату нечіткої логіки, а відповідні функції належності підлаштовуються з використанням алгоритмів навчання нейронних мереж. Для налаштування параметрів нечіткої системи програма *anfis* використовує метод зворотного розповсюдження помилки або гібридний алгоритм навчання [8,13].

Запис програми *anfis* має вигляд:

$$[fis, error] = anfis (trd, infis, trnopt), \quad (12)$$

де trd – навчальна вибірка у вигляді матриці, кожний рядок якої задає пару даних "входи – виходи"; infis – вихідна система нечіткого висновку Сугено нульового або першого порядку, створена за допомогою функції genfis1 або графічного інтерфейсу – FIS-редактора; trnopt – вектор параметрів налаштування: кількість ітерацій (10 – при замовчуванні), допустима помилка навчання (0 – при замовчуванні); fis – навчена система нечіткого висновку; error – помилка навчання, розрахована за формулою (9).

Слід зазначити, що навчання нечіткої системи Сугено за допомогою програми anfis у більшості випадків задовольняє точність ідентифікації об'єкта, необхідну при інженерних розрахунках.

Перед викликом функції anfis і заданням вхідних параметрів створюються і запам'ятовуються для кожної нечіткої підсистеми експертної системи (рисунок 3) навчальні вибірки: trdinfra, trdTurOb і trdKO. Записується в командному вікні послідовно вираз (12) для кожної з підсистем. Замість trd та infis записують імена раніше створених файлів: навчальної вибірки і підсистеми експертної системи, розробленої в графічному інтерфейсі. Після навчання отримуємо нову нечітку систему: infra_n.fis, TurOb_n.fis, KO_n.fis. Назви навчених систем записують у блоки Fuzzy Logic Controller (рисисунок 5). Після цього модель експертної системи готова до роботи. Похибка оцінки екологічної привабливості регіону в порівнянні з ненавченою експертною системою зменшилась більш ніж у 39 разів (таблиця).

Таблиця 1 – Похибки нечітких підсистем експертної оцінки екотуристичної привабливості регіону

Назва нечіткої підсистеми	Середньоквадратична похибка нечітких підсистем				
	Мамдані, створеної експертами	Мамдані, налаштованої функцією fmincon	Мамдані, налаштованої генетичним алгоритмом	Сугено, створеної експертами	Сугено, налаштованої програмою anfis
Інфраструкта	0.39789	0.0749	0.0601	0.16237	1.8e-008
Туристичні об'єкти	0.45318	0.0182	0.0117	0.10767	0.00272
Комплексна оцінка	4.5318	1.4807	0.9534	1.0767	0.02723

Навчання нечіткої системи Мамдані здійснюється за допомогою функцій пакету Optimization Toolbox або пакету Genetic Algorithm and Direct Search Toolbox середовища MatLAB. Для оптимізації нелінійних функцій з обмеженнями, застосовується функція fmincon пакету Optimization Toolbox.

Перед зверненням до цієї функції створюється і запам'ятовується М-файл ob_fun_infra [13]. Цей файл встановлює нові, визначені функцією fmincon, параметри X_{opt} нечіткої підсистеми infra і визначає середньоквадратичну похибку підсистеми. Задаються вектори: верхніх меж V_{lb_c} , нижніх меж V_{ub_c} і початкових значень S_0 коефіцієнтів концентрації термів вхідних і вихідної змінних; вектори координат максимуму функцій належності, відповідно, V_{lb_b} , b_0 , V_{ub_b} ; вектори ваг правил – V_{lb_w} , w_0 , V_{ub_w} . Нижні і верхні вектори меж об'єднуються у вектор V_{lb} , V_{ub} , а вектори початкових значень у вектор X_0 .

Звернення з командного рядка MatLAB при знаходженні оптимальних параметрів нечіткої системи infra виглядає наступним чином:

$$[Xopt, delta] = fmincon (@ob_fun_infra, X0, [], [], [], [], V_{lb}, V_{ub}, [], trdinfra (:, 1 : 2), trdinfra (:, 3)), \quad (13)$$

% trdinfra (:, 1 : 2) – перші два стовпчики навчальної матриці trdinfra (значення вхідних змінних); trdinfra (:, 3) – значення вихідних змінних.

% завантажують вхідну нечітку систему створену за допомогою графічного інтерфейсу

```

fis = readfis ('Infra');
fis = chage_fis_infra (Xopt, fis); % Розраховують похибку підсистеми
out_ch = evalfis (trdinfra (:, 1 : 2), fis);
RMG = norm (out_ch - trdinfra (:, 3)) / sqrt (length (trdinfra));
writefis (fis, Infra_n). % Записують навчену систему на диск
    
```

Таким же чином навчають підсистеми TurOb і КО. Похибки нечітких підсистем після навчання наведені в таблиці. Відкривається знову редактор FIS, операцією Import меню File завантажується навчена підсистема Infra_n. Активація блоків вхідних змінних дає можливість переглянути зміну форми і параметрів функцій належності, а виконання команди Rules меню Edit показує зміну ваг правил бази знань. На рисунку 5 наведені функції належності термів змінних infra після навчання.

Значення коефіцієнтів концентрації та центрів максимумів і форми функцій належності, а також ваги правил суттєво змінилися в порівнянні з тими значеннями, які вони мали до навчання (рисунок 2).

Навчання нечітких підсистем Мамдані за допомогою еволюційних обчислень здійснюється з командного рядка MatLAB за допомогою функції ga. Перед застосуванням функції необхідно створити m-файл цільової функції fitnessfun. Для нечітких підсистем цей файл створюється на основі залежності (8) і системи рівнянь (10).

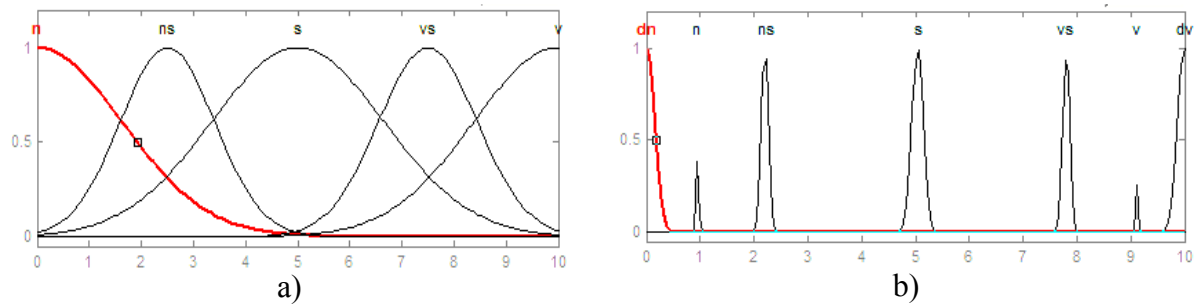


Рисунок 6 - Вигляд функцій належності вхідних і вихідної змінних підсистеми інфраструктури після навчання

Пошук мінімуму виразу (8) здійснювався зміною центрів максимуму функцій належності вхідних і вихідної змінних та ваг правил.

Програма пошуку мінімуму виразу (8) має вигляд:

```

nVars = 42; % задання кількості змінних
options = gaoptimset ;% задання початкових опцій алгоритму;
% задання меж зміни змінних. Перший стовпчик матриці відповідає
нижній межі, другий – верхній
options = gaoptimset (options, 'PopInttRange',
[x'(1), x'(2),...,x'(42);x''(1),x''(2), ...,x''(42)]);
% x'(1),...x'(17) – нижні, x''(1),...,x''(17) – верхні межі діапазону зміни
координат центрів функцій належності вхідних і вихідних
змінних; x'(18),...,x'(42) – нижні, x''(18),...,x''(42) – верхні межі
діапазону ваг правил бази знань системи Infra.
options = gaoptimset (options, 'PopulationSize',100); % Кількість популяцій
options = gaoptimset(options 'MutationFcn', {@mutationgaussian 1
1});% Характер мутацій
options = gaoptimset(options, 'Display', 'off');
% Розв'язок
[x, fval, reason, output, population, scores] = ga (fitnessFunction, nvars, options)
    
```

% x – вектор значень змінних, при яких вираз (8) досягає мінімуму; f_{val} – величина похибки; $reason$ – причина завершення виконання алгоритму; $population$ – кінцева популяція; $scores$ – значення цільової функції для особин кінцевої популяції.

Для визначення оптимальних параметрів підсистем TurOb і КО складаються програми, які відрізняються від наведеної вище кількістю змінних та діапазонів їх зміни.

Похибки нечітких підсистем Мамдані, навчених еволюційним алгоритмом, наведені в таблиці.

Результати досліджень показують, що найменшу похибку комплексної оцінки екотуристичної привабливості регіону має система нечіткого висновку Сугено. Крім цього, ця система простіша в реалізації і не потребує складних алгоритмів для налаштування. Тому, як модель експертної системи прийняття рішень при інвестуванні екотуризму, прийнята ієрархічна нечітка система Сугено. Похибка системи на тестовій вибірці не перевищує 1-2 %.

Висновки. Розроблена методика побудови інтелектуальних систем управління і прийняття рішень на основі інтеграції методів штучного інтелекту дозволяє ідентифікувати складні нелінійні об'єкти і процеси, формалізація яких складна або неможлива, в умовах неповної та неточної інформації. Дослідження процесу створення і якості функціонування нечіткої ієрархічної експертної системи оцінки привабливості регіону для екотуризму показали високу ефективність таких систем, особливо при їх реалізації на основі алгоритму Сугено, завдяки відносно простій програмі навчання та практичній реалізації.

У випадку, коли число змінних (параметрів налаштування нечітких систем) цільової функції складає двозначну величину, доцільно для її оптимізації застосувати еволюційні обчислення.

Література

1. Заде Л. Понятие лингвистической переменной и его применение к принятию приближённых решений. Пер. с англ. – М.: Мир, 1976.– 165с.
2. Терещин В.Н., Суховеева Н.И., Донец Л.Ю. Автоматизированные системы, основанные на знаниях // Адаптивні системи автоматичного управління.– Дніпропетровськ: Системи технології, 2002.– С. 118–123.
3. Ротштейн А.П. Интеллектуальные технологии идентификации: нечёткие множества, генетические алгоритмы, нейронные сети.– Винница: УНИВЕРСУМ –Винница, 1999.– 320с.
4. Круглов В.В., Борисов В.В. Искусственные нейронные сети. Теория и практика.– М.: Горячая линия – Телеком, 2001.– 382с.
5. Man K.F., Tang K.S., Kwong S. Genetic Algorithms–Concepts and Desings, Mathematics and Computers in Simulation, 2000, Vol51, s. 287–300.
6. Гладков Л.А., Курейчик В.В., Курейчик В.М. Генетические алгоритмы.– М.: ФИЗМАТЛИТ, 2006.– 320с.
7. Щерба А.А., Кирик В.В. Особенности программных средств для создания нечётких систем управления электротехническими устройствами. // Технічна електродинаміка.– 2007.– № 2.– С. 30–34.
8. Дьяконов В.П., Круглов В.В. MatLAB 6.5 SP1/7/7 SP1/7 + Simulink 5/6. Инструменты искусственного интеллекта и биоинформатики.– М.: СОЛОН–ПРЕСС, 2006.– 456с.
9. Старчкова Н.В. Этапы и механизмы оценки конкурентноспособности региональных рынков рекреационных услуг // Культура народов Причерноморья.– 2003.– № 45.– С. 79–84.
10. Леоненков А.В. Нечёткое моделирование в среде MatLAB и fuzzy TECH.– СПб: БХВ – Петербург, 2003.– 736с.
11. Корчемний М.О., Федорейко В.С. Інтелектуальні технології управління та прийняття рішень. Ч. 1. Розмита логіка.– Тернопіль: ТНПУ, 2007.– 140с.
12. Корчемний М.О., Скалига Л.М. Аналітичне проектування і практична реалізація інтелектуальних систем автоматичного керування // Вісник Тернопільського державного технічного університету.– 2007.– Том 12.– № 4.– С. 126–134.
13. Штовба С.Д. Проектирование нечётких систем средствами MatLAB.– М.: Горячая линия.– Телеком, 2007.– 288с.

Одержано 06.08.2008 р.