

**Д.І. Єгошкін, Н.А. Гук, С.Ф. Сірик**

*Дніпровський національний університет імені Олеся Гончара*

## **АЛГОРИТМ КЛАСИФІКАЦІЇ НА БАЗІ НЕЧІТКОЇ ЛОГІКИ З РОЗШИРЮВАНОЮ КІЛЬКІСТЮ ВИВОДІВ**

Розглядається задача автоматичної генерації бази знань, що складається з продукційних правил, для об'єктів навчальної вибірки з використанням методів нечіткої логіки і правила порівняння значень вихідних змінних. Пропонується алгоритм формування нечітких продукційних правил.

**Ключові слова:** штучний інтелект, нечітка логіка, відображення, класифікація, база знань, експертна система, мова C/C ++, мова JavaScript, JSON.

Рассматривается задача автоматической генерации базы знаний, состоящей из продукционных правил, для объектов обучающей выборки с использованием методов нечеткой логики и правила сравнения значений выходной переменной. Предлагается алгоритм формирования нечетких продукционных правил.

**Ключевые слова:** искусственный интеллект, нечеткая логика, отражение, классификация, база знаний, экспертная система, язык C / C ++, язык JavaScript, JSON.

In this article the problem of automatic generation of a knowledge base which consists of production rules for training dataset using fuzzy logic methods and a rule for comparing the values of an output variable is considered. An algorithm for the formation of fuzzy production rules is proposed. An actual problem of development and improvement of artificial intelligence algorithms and fuzzy logic application for solving a wider range of problems is considered. With the help of such systems are possible to eliminate the difficulties of formalizing knowledge about technological processes; also it is possible to organize recognition of nonstandard and emergency situations without using precise mathematical models and classical decision theory based on the tool of mathematical equations. The development of this area is relevant, as the number of tasks are constantly increasing, and the amount of knowledge becomes too large to handle them manually. The construction of an exact mathematical model for poorly formalized objects and processes are very difficult task, due to the lack of complete information. The situation becomes even more complicated if the properties of the object or process change dynamically. Therefore, the development of mathematical methods and algorithms that allow structuring the system of rules and determining the order of their calls to control consistency and completeness to optimize the number of rules, are an actual task.

Modern approaches to the automation of these processes are considered. These approaches significantly improve the work of expert systems, but they allow to work only with static knowledge bases, limit the number of logical inferences and are not applicable for cases when it is necessary to add new logical rules to the existing system. In this article, an approach is developed that makes it possible to expand the knowledge base of the expert system with new rules in the process of exploitation. The developed algorithm has following advantages: high speed of problem solving; the ability that allows expanding the number of system responses without changing the scope of the rules and the program itself; expanding the range of application of fuzzy logic algorithms.

The developed algorithm has following disadvantages: if the system's response database has objects that are similar to each other, they can have the same center of gravity, which in turn leads to additional checks; the minimum distance for mapping the object should be selected experimentally. The application of this algorithm can be seen on the website of the program, which classifies, maps an arbitrary user in a set of comic book characters database "CMD - Combat Marvel DC" [8]. The approach that was proposed has been successfully implemented using the C/C++ and JavaScript languages, and JSON open-standard file format that uses human-readable text to transmit data objects consisting of attribute-value pairs and array data types. Software that was used for development: NetBeans IDE, MinGW, GNU Compiler Collection, WhiteStarUML, GitHub, WebGL, Chrome, Mozilla Firefox, Opera.

**Keywords:** artificial neural network, error back-propagation method, anthropological points, oriented gradient histogram, convolutional neural network, prediction.

**Вступ.** Розвиток обчислювальних технологій і апаратної частини комп'ютера дозволив широко застосовувати експертні системи для підтримки прийняття рішень в таких галузях знань, як економіка, промисловість, медицина. Сформульовані бази знань і розроблені алгоритми логічного виведення дозволяють акумулювати великий обсяг знань, отриманих від експертів. За допомогою таких систем можна усувати труднощі формалізації знань про технологічні процеси, організувати розпізнавання нестандартних та аварійних ситуацій без використання точних математичних моделей, які базуються на апараті математичних рівнянь і класичної теорії прийняття рішень.

Побудова точної математичної моделі для погано формалізованих об'єктів і процесів є досить складною задачею у зв'язку з відсутністю повної інформації. Ситуація ще більше ускладнюється, якщо властивості об'єкта або процесу змінюються динамічно. Крім того, знання є такою структурою, що постійно змінюється і розвивається, а це в свою чергу може призвести до необхідності повної переробки математичної моделі.

Слід зазначити, що якість функціонування експертної системи залежить від повноти, несуперечності, а також розміру бази знань. Велика кількість правил призводить як до зниження швидкості виконання логічного виводу, що неприпустимо для систем, які працюють у режимі реального часу, так і до суперечливості накопичених знань.

Тому розробка математичних методів і алгоритмів, що дозволяють структурувати систему правил і визначати порядок їх викликів, контролювати несуперечність і повноту, оптимізувати кількість правил, є актуальною задачею. При формуванні баз знань експертних систем існує 2 підходи: аналіз знань експертом (групою експертів) на основі досвіду, або автоматичне формування бази знань з використанням методів інтелектуального аналізу даних і алгоритмів машинного навчання. Використання другого підходу дозволяє виконувати процедури створення і контролю баз знань в автоматичному режимі. В даний час існує декілька підходів до автоматизації зазначених процесів.

В [1] з використанням теорії графів продукційна база знань представляється у вигляді мультиграфа, в якому кожному продукційному правилу відповідає власний підграф. Побудована таким чином база знань структурується, для збільшення швидкодії механізму логічного виводу умови правил зв'язуються зі значеннями атрибутів в робочій пам'яті, що дозволяє врахувати вплив результатів виконання одних продукційних правил на умови реалізації інших. В [2] розроблено математичну модель, яка представляє собою гіперграф. Всі сутності і залежності, що представлені в базі знань, об'єднуються у гіперграфі. Модифіковані алгоритми прямого та зворотного виведення здійснюють пошук на отриманому підграфі. В роботі [3] запропоновано підхід до структурування бази знань шляхом визначення рейтингів правил з подальшим видаленням суперечливих правил і правил з найменшими рейтингами.

Методи інтелектуального аналізу даних і нейронечітких технології останнім часом знаходять застосування при створенні та аналізі баз знань. Так в [4] запропонована математична модель нейронної мережі, яка допускає пряме перетворення бази знань в мережу. З використанням генетичного алгоритму виконується параметрична оптимізація структури бази знань. В роботі [5] пропонується поєднання методу кластерного аналізу та нечіткої моделі логічного виводу Такагі-Сугено для редукції бази знань, схожі правила логічного виведення об'єднуються в один кластер. Для оцінки схожості передумов правил при однакових значеннях логічних висновків використовуються спеціальні метрики. На число логічних висновків накладаються кількісні обмеження. В роботі [6] задача кластеризації знань в системах штучного інтелекту розв'язується із застосуванням мурашиних алгоритмів.

Зазначені підходи істотно поліпшують роботу експертних систем, однак дозволяють працювати тільки зі статичними базами знань, накладають обмеження на кількість логічних висновків і не можуть бути застосовані для випадків, коли в існуючу систему необхідно додавати нові логічні правила.

У даній роботі розвивається підхід, який дозволяє розширювати базу знань експертної системи новими правилами в процесі експлуатації.

**Постановка задачі.** Припустимо, що існує сформована продукційна база знань  $R = \{R_1, R_2, \dots, R_N\}$ , де  $R_i$  – нечітке продукційне правило;  $i = \overline{1, N}$ ;  $N$  – вихідна кількість правил в базі знань. При цьому передбачається, що у множені правил виділені  $k$  підмножин ( $k < N$ ), у яких правила згруповані за результатом логічного висновку. Підмножина правил містить правила, з використанням яких описуються приблизно однакові нечіткі закономірності в аналізованих даних і виконується класифікація об'єктів за допомогою алгоритму нечіткого логічного виведення. Передбачається, що існуюча система правил може бути розширена, коли у систему потрапляє новий об'єкт.

При цьому структура правила системи залишається незмінною. Подібні вимоги дозволяють уникати додаткових перевірок логічних правил на наявність протиріч після додавання нових правил в систему.

Необхідно розробити підхід, з використанням якого при виконанні процедури прямого логічного виведення стає можливим формування і додавання в систему нових продукційних правил для класифікації об'єктів, параметри яких відрізняються від вже використаних у навчальній вибірці при складанні бази правил.

**Продукційна модель представлення знань.** Розглядається задача ідентифікації об'єктів з використанням алгоритму нечіткого логічного виведення та системи продукційних правил.

Об'єкт  $X$  характеризується вектором параметрів  $(x_1, x_2, \dots, x_n)$ , де  $x_i$  - вхідна лінгвістична змінна. З кожною лінгвістичною змінною пов'язано її нечітке значення  $A_j$ .

З кожним об'єктом  $X$ , що характеризується набором вхідних параметрів, зв'язується вихідна змінна  $Y$ . Вхід і вихід об'єкту, що досліджується, пов'язані між собою функціональною залежністю виду:

$$Y = f(x_1, x_2, \dots, x_n) \quad (1)$$

де  $f$  – дійсна функція від чітких значень  $(x_1, x_2, \dots, x_n)$

Функція  $f(x_1, x_2, \dots, x_n)$  набуває дискретних значень, оскільки число різних заключень логічних правил при будь-яких значеннях аргументів  $x_1, x_2, \dots, x_n$  є скінченним.

При розв'язанні задачі класифікації значеннями функції  $f(x_1, x_2, \dots, x_n)$  є константи, що вказують на клас до якого належить об'єкт.

Апріорну інформацію стосовно залежності (1) можна зобразити у вигляді сукупності продукційних правил у такий спосіб:

$$\begin{aligned} P_p : \text{Якщо } x_1 \in A_{p1} \wedge x_2 \in A_{p2} \wedge \dots \wedge \\ x_n \in A_{pn} \text{ ТО } Y = Y_p, \end{aligned} \quad (2)$$

де  $p = \overline{1, P}$  – номер правила в базі правил;  $P$  – загальна кількість правил;  $A_{pj}$  – нечітке значення змінної  $x_i$  в термі  $j$ ,  $i = \overline{1, N}$ ;  $Y_p$  – мітка деякого класу, до якого належить об'єкт  $X$ .

Сформована база правил може бути розбита на систему підмножин, що перетинаються, за значенням мітки класу в логічному заключенні правила:

$$R = \{R_1, R_2, \dots, R_N\} = \{Rul_1 | Y = Y_1 \cup Rul_2 | Y = Y_2 \cup \dots \cup Rul_k | Y = Y_k\}$$

де  $Rul_k$  – підмножина правил, в яких  $Y = Y_k$ . Для відображення чітких вхідних значень  $x_j$  – в нечіткі множини вводяться функції приналежності  $M_{pj}$  виду:

$$M_{pj}(x_j, a_p, b_p, c_p, d_p) = \begin{cases} 0, & x_j \leq a_p \\ \frac{x_j - a_p}{b_p - a_p}, & a_p \leq x_j \leq b_p \\ 1, & b_p \leq x_j \leq c_p \\ \frac{d_p - x_j}{d_p - c_p}, & c_p \leq x_j \leq d_p \\ 0, & d_p \leq x_j \end{cases} \quad (3)$$

де  $a_p, b_p, c_p, d_p$  - числові параметри, які визначають границі термів, набувають дійсних значень і впорядковані відношенням:  $a_p \leq b_p \leq c_p \leq d_p$ .

Границі термів визначаються за формулами відносно бази даних навчальної вибірки:

$$\bar{x}_j = \begin{cases} \left( \min_{i=1} (x_i), \max_{i=1} (x_i) \right), & j=1, J=2 \\ \left( \min_{i=1} (x_i), \max_{i=1} (x_i) \right), & j=J, J=2 \\ \left( \min_{i=1} (x_i), \min_{i=1} (x_i) + \frac{\left( \text{avg}_{i=1} (x_i) \right)}{J-2} \right), & j=1 \\ \left( \min_{i=1} (x_i) + \frac{(j-1) \cdot \text{avg}_{i=1} (x_i)}{J-1}, \max_{i=1} (x_i) - \frac{(J-j) \cdot \text{avg}_{i=1} (x_i)}{J-1} \right), & 1 < j < J \\ \left( \max_{i=1} (x_i) - \frac{\left( \text{avg}_{i=1} (x_i) \right)}{J-2}, \max_{i=1} (x_i) \right), & j=J \end{cases}, \quad (4)$$

де  $j=1, J, J \geq 2, J$  - кількість елементів терм-множини лінгвістичної змінної, якщо  $J < 2$  - немає сенсу створювати таку лінгвістичну змінну.

**Метод розв'язання.** Для виконання процедури нечіткого логічного виведення введемо такі операції, аналогічно [8].

Для виконання процедури фазифікації вхідних змінних  $x_i$  в вектор нечітких множин  $A_{pj}$  будемо використовувати наступну операцію:

$$A_{pj} = \int_{\underline{x}_i}^{\overline{x}_i} (M_{pj}(x_i)/x_i) dx \quad (5)$$

де  $\underline{x}_i, \overline{x}_i$  – границі термів вхідних змінних.

Для виконання процедури дефазифікації будемо використовувати:

$$Y_p = \int_{\underline{Y}}^{\overline{Y}} (M_{Y_p}(Y)/Y) dY, Y \in [\underline{Y}; \overline{Y}] \quad (6)$$

де  $\underline{Y}, \overline{Y}$  – границі терму вихідної змінної.

Ступінь приналежності вхідного об'єкта  $X^* = (x_1^*, x_2^*, \dots, x_n^*)$  нечітким термам  $Y_p$  з бази знань (2) описується системою нечітких логічних рівнянь:

$$M_{Y_p}(x^*) = \bigvee_{p=1, P_i=1, N} \bigwedge [M_{pj}(x_i^*)], j = \overline{1, m} \quad (7)$$

де оператори  $\vee$  та  $\wedge$  відповідають виконанню логічних операцій «АБО» та «І» відповідно. В роботі використані їх реалізації у вигляді знаходження *max* та *min*.

Нечітка множина  $\tilde{Y}^*$ , що відповідає вхідному вектору  $X^*$  визначається у вигляді:

$$\tilde{Y}^* = \mathit{agg}_{j=1, m} \left( \int_{\underline{Y}}^{\overline{Y}} \mathit{imp}(M_{Y_p}(X^*), M_{Y_p}(Y)/Y) dY \right) \quad (8)$$

де *imp* – операція імплікації, *agg* – операція агрегування, які реалізовані операцією знаходження *min* та *max* відповідно.

Чітке значення виходу  $Y^*$  визначається в результаті дефазифікації нечіткої множини  $\tilde{Y}^*$  за методом центру тяжіння:

$$Y^* = \int_{\underline{Y}}^{\overline{Y}} Y \cdot M_{\tilde{Y}^*}(Y) dY / \int_{\underline{Y}}^{\overline{Y}} M_{\tilde{Y}^*}(Y) dY \quad (9)$$

Для виконання ідентифікації об'єкта використовується алгоритм нечіткого логічного виведення [7], модифікований введенням додаткового механізму порівняння чіткого вихідного значення  $Y^*$ , отриманого для об'єкта  $X^* = (x_1^*, x_2^*, \dots, x_n^*)$ , і вихідних значень об'єктів  $Y_i$  наявних в базі знань. Схему модифікованого алгоритму нечіткої логіки наведено на рис. 1. Відстань між об'єктами визначається на основі обраної метрики в просторі характеристик. Для оцінки міри близькості елементів використовується Евклідова відстань:

$$d(Y^*, Y_i) = \sqrt{\sum_{t=1}^m (Y_t^* - Y_{it})^2} \quad (10)$$

$$d(Y^*, Y_i) < \varepsilon \quad (11)$$

У разі, коли для об'єкта  $X^*$  умова (11) не виконується, необхідно сформулювати нове логічне правило:

$$P_{p+1} : \text{Якщо } x_1 \in A_{k1} \wedge x_2 \in A_{k2} \wedge \dots \wedge x_n \in A_{kn} \text{ ТО } Y_{p+1} = X^*, \quad (12)$$

де  $A_{kj}$  – нечіткі значення змінних  $x_j$ , які обчислюються з використанням функції приналежності (5) при значеннях  $a_{k1} = x_1^*, a_{k2} = x_2^*, \dots, a_{kn} = x_n^*$ .

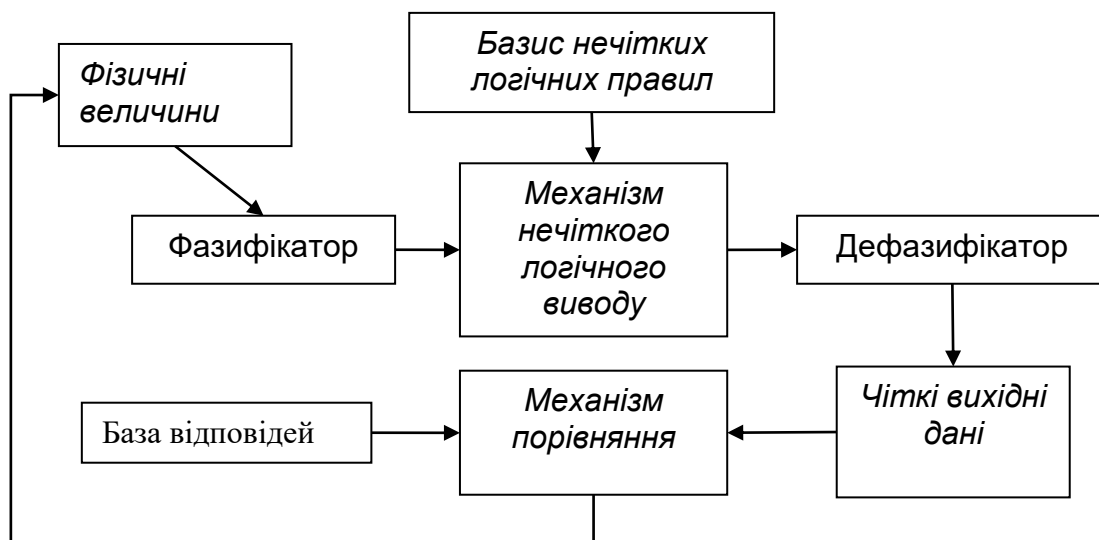


Рис. 1. Модифікований алгоритм нечіткої логіки

У разі, коли для об'єкта  $X^*$  умова (11) не виконується, необхідно сформулювати нове логічне правило:

$$P_{p+1} : \text{Якщо } x_1 \in A_{k1} \wedge x_2 \in A_{k2} \wedge \dots \wedge x_n \in A_{kn} \text{ ТО } Y_{p+1} = X^*, \quad (12)$$

де  $A_{kj}$  – нечіткі значення змінних  $x_j$ , які обчислюються з використанням функції приналежності (5) при значеннях  $a_{k1} = x_1^*, a_{k2} = x_2^*, \dots, a_{kn} = x_n^*$ .

Після виконання процедури дефазифікації обчислюється  $d(Y^*, Y)$  та перевіряється умова (11).

Вказану послідовність дій для існуючої бази правил можна описати наступним алгоритмом.

Алгоритм:

Крок 1. Задати значення  $\varepsilon$  – похибка системи,  $P$  – загальне число правил, ініціалізувати вхідні змінні, задати функції приналежності.

Крок 2. Для об'єкта  $X^*$  на базі сформованих продукційних правил за допомогою операцій алгоритму нечіткого логічного виведення (5) - (8) розрахувати ступені приналежності об'єкта чітким множинам  $\tilde{Y}_p$ .

За формулою (9) отримати чітке значення  $Y^*$  для об'єкта  $X^*$ .

Для об'єктів  $Y_k$ , які є логічними висновками правил з системи (2), так само виконати процедуру нечіткого логічного виведення та отримати відповідні значення  $Y_k^*, k = \overline{1, K}$ .

Крок 3. Визначити відстані між об'єктом  $Y^*$  і існуючими об'єктами  $Y_k$ , перевірити виконання умови (11).

Якщо нерівність є вірною перейти до кроку 6, інакше перейти до кроку 4.

Крок 4. Доповнити базу знань правилом (12);  $p := p + 1$ .

Крок 5. Перерахувати границі термів нечітких множин відповідно до формул (4).

Крок 6. Видати результат ідентифікації.

Крок 7. Перевірка правил останову:

Перевірити, чи всі об'єкти навчальної вибірки переглянуті, якщо так, то зупинитися, інакше обрати наступний об'єкт і перейти до кроку 3.

**Аналіз результатів.** Запропонований підхід до формування бази правил при пред'явленні елементів навчальної вибірки був протестований для задачі класифікації користувача відносно персонажа бази даних «CMD - Combat Marvel DC»[8]. В якості вхідних параметрів персонажа виступають: зріст (см), вага (кг), вік (роки). Зазначені параметри описуються нечіткими значеннями.

Формуються відповідні терми вхідних змінних:

Зріст (низький, середній, високий);

Вага (низька, середня, висока);

Вік (молодий, середній, дорослий).

Вихідна змінна характеризує класи об'єктів: Клас (I, II, III).

Для кожної вхідної та вихідної змінних вводяться функції приналежності виду (3), обчислюються границі термів за формулами (4).

Границі термів:

Зріст\_низький = (155, 174.6);

Зріст\_середній = (164.83, 186.33);

Зріст\_високий = (174.6, 198);

Вага\_низка = (50, 73.33);

Вага\_середня = (61.6, 89.16);

Вага\_висока = (73.33, 105);

Вік\_молодий = (19, 26.33);

Вік\_середній = (22.7, 30.7);

Вік\_дорослий = (26.33, 35);

Клас\_I = (0, 0.5);

Клас\_II = (0.25, 0.75);

Клас\_III = (0.5, 1);

Формується система правил, передумови яких складені з усіх можливих комбінацій значень нечітких вхідних змінних (всього 27 правил).

У таблиці представлені результати ідентифікації різних об'єктів при заданих значеннях  $\varepsilon = 2,5$ ,  $P = 27$ , база даних персонажів:  $X_{01}(198,105,35)$ ,



$X_{02}(171,65,25)$ ,  $X_{03}(155,50,19)$ . Вхідні об'єкти:  $X_1(175,70,20)$ ,  $X_2(180,73,21)$ . У таблиці 1 подано результати ідентифікації.

Для об'єкта  $X_1$  система повертає об'єкт  $X_{02}$ , який знаходиться найближче. Фактично ми відобразили об'єкт  $X_1$  в множину об'єктів бази відповідей  $X_{01}, X_{02}, X_{03}$ . Для об'єкта  $X_2$  відстань  $d = 8,072$ , тобто умова  $d < \varepsilon$  не виконується, отже, потрібно додати цей об'єкт і розширити систему правил правилом з новим заключенням.

Таблиця 1

## Результати ідентифікації

	Вхідні параметри			Дефазфіцировані значення $Y^*$	$\min_{j=1} (d(X_i, X_j))$	Результат класифікації $i$	Заклучення
	Зріст (см)	Вага (кг)	Вік (роки)				
$X_1$	175	70	20	42,857	2,179	Клас_II	X02
$X_2$	180	73	21	48,75	8,072	Клас_II	Новий
$X_{01}$	198	105	35	75	0	Клас_I	X01
$X_{02}$	171	65	25	40,678	0	Клас_II	X02
$X_{03}$	155	50	19	25	0	Клас_III	X03

Запропонований алгоритм, було реалізовано у вигляді програмного продукту з використанням мов C/C++ та JavaScript, а також текстового формату обміну даними JSON. Для розробки використовувалися: NetBeans IDE, WhiteStarUML, GitHub, WebGL, Chrome, Mozilla Firefox, Opera.

Розроблений алгоритм має наступні переваги: висока швидкість розв'язання задачі; можливість розширення кількості відповідей системи, без зміни існуючих правил у базі знань і алгоритму логічного виведення; розширення спектру застосування алгоритмів нечіткої логіки.

Розроблений алгоритм має наступні недоліки: якщо в базі відповідей системи є об'єкти схожі один на одного, вони можуть мати однаковий центр ваги, що в свою чергу призводить до додаткових перевірок; мінімальну відстань для відображення об'єкта потрібно підбирати експериментальним шляхом.

**Висновки.** Запропоновано підхід до автоматичної генерації продукційних правил бази знань на основі порівняння нових об'єктів з вже існуючими у системі за допомогою метрики, модифіковано алгоритм нечіткого логічного виведення шляхом додавання блоку порівняння вихідного значення для нового об'єкту з вихідними значеннями для наявних у базі знань об'єктів.

Подальші дослідження будуть спрямовані на вдосконалення запропонованого підходу шляхом обчислення чітких вихідних даних для бази відповідей заздалегідь і внесення цих даних до бази відповідей, але для цього буде потрібно створити механізм контролю даних для постійного їх

оновлення і перезапису в разі зміни або модифікації правил системи, а також в разі додавання нових відповідей системи.

### Бібліографічні посилання

1. **Домнич В.С., Иващенко В.А.** Построение базы знаний для поиска причин аварийных ситуаций при формировании листового стекла / В.С. Домнич, В.А. Иващенко // УБС. – 2011. – Вып. № 33. – С. 218–232.
2. **Иванов А.С.** Модель представления продукционных баз знаний на ЭВМ / А.С. Иванов // Изв. Саратов. ун-та. Нов. сер. Сер. Математика. Механика. Информатика. –2007. – 7:1. – С. 83–88.
3. **Сергиенко М.А.** Методы проектирования нечеткой базы знаний / М.А. Сергиенко // Вест. Воронеж. гос. ун-та. Серия: Системный анализ и информационные технологии. – 2008. – № 2. – С. 67–71.
4. **Бухнин А.В., Бажанов Ю.С.** Оптимизация баз знаний экспертных систем с применением нейронных нечетких сетей / А.В. Бухнин, Ю.С. Бажанов // Нейрокомпьютеры: разработка, применение. –2007. – №11.
5. **Абдулхаков А.Р., Катасёв А.С.** Кластерно-генетический метод редукции баз знаний интеллектуальных систем / А.Р. Абдулхаков, А.С. Катасёв // Фундаментальные исследования. – 2015. – № 5-3. – С. 471–475.
6. **Щуревич Е.В.** Кластеризация знаний в системах искусственного интеллекта / Е.В. Щуревич // Информационные технологии. – 2009. – №2. – С. 25–29.
7. **Ротштейн А.П.** "Интеллектуальные технологии идентификации: нечеткая логика, генетические алгоритмы, нейронные сети". – [Электронный ресурс]. – Режим доступа: <http://matlab.exponenta.ru/fuzzylogic/book5/index.php>
8. **Егошкин Д.И.** "CMD - Combat Marvel DC". – [Электронный ресурс]. – Режим доступа: <https://knightdanila.github.io/CMD/index.html>

*Надійшла до редколегії 15.09.2018.*