

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра комп'ютерних наук

Пояснювальна записка

до магістерської кваліфікаційної роботи
на тему:

Інформаційна технологія управління біоконверсією

Виконав: студент 2 курсу, групи 1КН-18м
Спеціальності 122
«Комп'ютерні науки»

Шалак А.В.
(прізвище та ініціали)

Керівник к.т.н., доц. Сілагін О. В.
(прізвище та ініціали)

Рецензент к.т.н., доц.каф.ПЗ Войтко В.В.
(прізвище та ініціали)

Вінниця - 2019 року

АНОТАЦІЯ

В магістерській кваліфікаційній роботі “Інформаційна технологія управління біоконверсією” досліджується можливість застосування апарату нечіткої логіки до задачі оптимізації управління реактором для виробництва біогазу. Використання експертно-лінгвістичних закономірностей, що формалізуються засобами нечіткої логіки, дозволяє збільшити об’єм і підвищити якість виробленого біогазу. Результати роботи можуть бути використані для широкого кола задач пов’язаних з управлінням біохімічними технологічними процесами.

Для розробки модуля використанні програмні середовища: Visual Studio та FUZZY EXPERIENCE.

ABSTRACT

The master's qualification work "Bioconversion Management Information Technology" explores the possibility of applying the fuzzy logic apparatus to the task of optimizing the reactor management for biogas production. The results of the work can be used for a wide range of tasks related to the management of biochemical technological processes.

To develop the module use the following software environments: Visual Studio and FUZZY EXPERIENCE.

ЗМІСТ

ВСТУП.....	3
1. ОБГРУНТУВАННЯ ДОЦІЛЬНОСТІ МОДЕРНІЗАЦІЇ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ.....	9
1.1 Біоконверсія та фактори, що впливають на неї.....	9
1.2. Багатофакторний аналіз технологічного процесу біоконверсії.....	11
1.3 Аналіз відомих рішень в управлінні технологічними процесами.....	13
1.4 Постановка задачі.....	17
1.5 Висновок.....	19
2. МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ УПРАВЛІННЯ БІОКОНВЕРСІЄЮ З ВИКОРИСТАННЯМ НЕЧІТКОЇ ЛОГІКИ	21
2.1 Обґрунтування вибору методології для моделювання роботи інформаційної технології.....	21
2.1.1 Нечіткі множини.....	22
2.1.2 Нейронні мережі.....	36
2.1.3 Генетичний алгоритм.....	49
2.2 Етапи розробки інформаційної технології.....	50
2.3 Дерево виведення рішень.....	52
2.4 Функції належності вхідних лінгвістичних термів.....	54
2.5 Моделювання та розробка бази знань.....	63
2.6 Висновок.....	67
3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ УПРАВЛІННЯ БІОКОНВЕРСІЄЮ.	68
3.1 Обґрунтування та вибір середовища та мови програмування.....	68
3.2 Декомпозиція програмного забезпечення та розробка структури компонентів.....	69
3.3 Розробка діаграми класів модуля виводу.....	72
3.4 Розробка алгоритма логічного виведення результату.....	75
3.5 Реалізація компонентів модуля з використанням екранних форм	

середовища FUZZY EXPERIENCE.....	77
3.6 Висновок.....	79
4. ЕКОНОМІЧНА ЧАСТИНА.....	80
ВИСНОВКИ.....	92
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	94
ДОДАТОК А.....	96
ДОДАТОК Б.....	107
ДОДАТОК В.....	129

ВСТУП

Актуальність роботи. Біоконверсія - це перетворення біомаси та побутових відходів за допомогою мікроорганізмів з метою отримання теплової енергії та біопалива. Іншими словами це перетворення мікроорганізмами одних речовин в інші. Конверсія широко використовується у всьому світі для отримання біогазу. У 2013 році біогазових установок, тільки в самому Китаї, налічувалося до 27 млн., Що дало можливість забезпечити паливом жителів сільської місцевості.

На сьогоднішній день біоконверсія - це провідна галузь біотехнологій. Вона розвивається швидкими темпами. За допомогою цього методу перетворення, щодня з біомаси виходить 6000т бутанола і 4000т суміші етанолу з ацетоном. Також біоконверсія можна використовувати в комплексі з іншими методами перетворення (електроліз, термохімічний метод і т.д.). Це свідчить про те, що тема магістерської кваліфікаційної роботи є актуальною.

Зв'язок роботи з науковими програмами, планами, темами. Магістерська робота виконана відповідно до напрямку наукових досліджень кафедри комп'ютерних наук Вінницького національного технічного університету 22 К1 «Моделі, методи, технології та пристрої інтелектуальних інформаційних систем управління, економіки, навчання та комунікацій» та плану наукової та навчально-методичної роботи кафедри. Ця робота виконується також в рамках співробітництва між кафедрою КН ВНТУ та ТОВ «ІТІ», а результати плануються до впровадження на ТОВ «ІТІ»

Мета та завдання дослідження. Метою дослідження є підвищення рівня виходу біогазу в процесі біоконверсії.

Для досягнення поставленої мети потрібно виконати наступні задачі:

- обґрунтувати доцільність доопрацювання інформаційної технології управління біоконверсією;

- проаналізувати існуючі технології, методи і моделі управління біоконверсією;
- сформулювати вимоги до роботи технології та розробити ТЗ;
- доопрацювати існуючу інформаційну технологію управління біоконверсією з врахуванням додаткових факторів, таких як стабільність температурного режиму та якість додаткового обладнання;
- провести математичне моделювання роботи етапів доопрацьованої технології з використанням апарату нечіткої логіки;
- провести фазифікацію розробленої моделі;
- розробити та наповнити базу знань у вигляді матриць з правилами ЯКЦО-ТО;
- на основі доопрацьованої технології виконати проектування програмного модуля управління біоконверсією;
- реалізувати та налаштувати роботу програмного модуля управління біоконверсією;
- протестувати роботу налаштованого програмного модуля;
- виконати задачі економічного розділу.

Об’єктом дослідження є процес управління біоконверсією

Предметом є інтелектуальні технології управління біоконверсією, математичні моделі, алгоритми та програмні засоби автоматизації управління біоконверсією

Як методи дослідження в роботі застосовано теорію нечітких множин (нечіткі рівняння, нечітка логіка), продукційна база знань, нечітке логічне виведення.

Наукова новизна одержаних результатів полягає в наступному:

- доопрацьовано існуючу інформаційну технологію управління біоконверсією, яка відрізняється від існуючої технології тим, що враховує додаткові фактори впливу на процес біоконверсії, а саме:

стабільність температурного режиму та якість додаткового обладнання, чим збільшує вихід біогазу в процесі біоконверсії;

- розроблена та фазифікована спеціалізована математична модель роботи доопрацьованої технології управління біоконверсією.

Практичне значення одержаних результатів полягає у наступному:

- розроблено алгоритм нечіткого логічного управління процесом біоконверсії;
- розроблено алгоритм нечіткої логічної апроксимації;
- реалізовано експериментальний програмний засіб.

Достовірність теоретичних положень магістерської кваліфікаційної роботи підтверджується строгістю постановки задач, коректним застосуванням математичних методів під час доведення наукових положень, строгим виведенням аналітичних співвідношень, порівнянням результатів, отриманих за допомогою розроблених у роботі методів, з відомими, та збіжністю результатів математичного моделювання з результатами, що отримані під час тестування розроблених програмних засобів.

Особистий внесок магістранта. Усі результати отримано автором самостійно. У працях, опублікованих у співавторстві, магістранту належать:
[1] – доопрацьована інформаційна технологія управління біоконверсією та спеціалізована математична модель управління біоконверсією, що базується на використанні середовище моделювання засобами нечіткої логіки.

Апробація результатів роботи. Результати досліджень апробовано в доповіді на НПК «МОЛОДЬ В НАУЦІ: ДОСЛІДЖЕННЯ, ПРОБЛЕМИ, ПЕРСПЕКТИВИ – 2020.

Публікації. За результатами магістерської кваліфікаційної роботи опубліковано: тези доповіді конференції [1].

Результати, одержані в процесі виконання магістерської кваліфікаційної роботи, плануються до впровадження в розробки науково-виробничого підприємства ТОВ «ІТІ».

1. ОБГРУНТУВАННЯ ДОЦІЛЬНОСТІ СТВОРЕННЯ НОВОЇ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ

1.1 Біоконверсія та фактори, що впливають на неї.

Біоконверсія - це найбільш перспективний, реальний і економічний спосіб переробки біомаси в паливо. Це мікробіологічне отримання палива (анаеробний процес) за допомогою мікроорганізмів, біохімічна суть якого полягає в скиданні електронів відмінними від кисню акцепторами з утворенням відновлювальних речовин: спиртів, кетон, сірководень, водню і метану.

Ефективність запасу енергії в водні при бродінні - 0 ... 30% від вихідного палива, а в метан переходить більше 80% енергії спочатку закладеної в вихідної органіці. Практично будь-яка органічна сировина може бути піддана метановому бродінню. Найбільш важлива сировина для отримання біогазу - це вторинна біомаса (відходи, які підлягають очистці та знищення), комунальні стоки міст, відходи мікробіологічної, масломолочної та інших галузей промисловості, відходи рослинництва і тваринництва. В даний час випускається велика кількість типів установок біогазогенераторов [2].

Розрізняють два види протікання біоконверсії:

1. Одноступінчаста біоконверсія включає в себе процес бродіння (в результаті чого утворюються спирти і кислоти органічні), ізомеризація (отримання фруктози за допомогою ферментів з глюкози) і утворення стероїдних гормонів (використовується у фармацевтиці для отримання фармпрепаратів).
2. Багатоступенева біоконверсія [3] відбувається в кілька етапів поки досягається потрібний результат. Для реалізації багатоступеневої біоконверсії потрібно використовувати змішані культури мікроорганізмів або ж додавання на різних стадіях процесу різні штами мікроорганізмів. До багатоступінчастої біоконверсії можна віднести процеси очищення стічних вод, отримання кормових білків, природну біоконверсію і отримання біологічно активних

речовин (гормони, вітаміни і антибіотики, використовуються в фармацевтиці та медицині).

Найбільшого поширення набула технологія аеробної і анаеробної конверсії [4]. Ці технології широко використовуються для переробки відходів сільського господарства з отриманням продуктів багатих біологічно активними речовинами.

Розвиток біоконверсії складається з виведення нових штамів мікроорганізмів, підвищення їх життєвого циклу, а також використання нових організмів, таких як дощові черв'яки і виноградні равлики.

У порівнянні з хімічними способами перетворення біоконверсія має ряд переваг:

- Продукти, які виходять під час цього процесу, мають чіткий склад і не містять домішок в кінцевому продукті.
- Цей процес набагато дешевше і менш витратний, так як речовини, за допомогою яких відбувається процес біоконверсії, синтезуються самими мікроорганізмами.
- Як правило, процеси, що реалізуються за допомогою цієї технології, проходять з реакцією одного типу.
- В результаті перетворення виходять продукти, що мають родинні структури.
- Процес біоконверсії проходить при м'яких умовах (низька температура і малих витратах енергії, при цьому швидкість реакції досить висока).
- В процесі очищення різних забруднень не утворюються інші шкідливі речовини.

Біоконверсія використовується в багатьох галузях. Використання в фармацевтиці та медицині дає можливість отримати медичні препарати, такі як гормони, інсулін та інші. Використання в харчовій промисловості дає можливість отримати клітинні дріжджі, гриби, білки харчові та кормові. За допомогою біоконверсії відбувається очищення стічних вод і утилізація

побутових органічних відходів, що в свою чергу благотворно впливає на екологію.

1.2. Багатофакторний аналіз технологічного процесу біоконверсії

Технологічний процес біоконверсії (ТПБК) - це процес при виконанні якого здійснюється перетворення відходів сільськогосподарського виробництва в біогаз. Впровадження таких процесів забезпечує можливість вирішення двох актуальних проблем, екологічної - очищення сільськогосподарських територій від забруднюючих відходів; енергетичної - отримання біогазу, який може використовуватися як корисне паливо.

Найважливішим критерієм якості функціонування ТПБК є кількість отриманого на виході біогазу. Тому прогнозування вихідної кількості біогазу в залежності від факторів, що впливають є необхідним елементом проектування пристроїв і процесів біоконверсії.

Традиційний підхід до моделювання багатофакторних технологічних процесів спирається на математичну теорію планування експерименту [3]. Застосування цієї теорії передбачає можливість реєстрації вихідного показника (в даному випадку - кількості біогазу) при різних значеннях факторів, що впливають з подальшим побудовою регресійної залежності. Однак відсутність експериментальної установки, яка існує в проекті, позбавляє дослідника такої можливості. В цьому випадку єдиним джерелом інформації для моделювання є експертні оцінки, які мають якісний характер.

Нижче розглядається інформаційна технологія яка базується на моделях нечіткої ідентифікації А.П.Ротштейна і деякі результати моделювання ТПБК з метою покращення цієї технології [10]. Основою моделювання є експертна лінгвістична інформація, яка доступна проектувальнику.

Оскільки ТПБК є типовим представником широкого класу хіміко-технологічних процесів, то доопрацьована технологія може бути використана

для багатофакторного аналізу таких процесів на ранніх стадіях їх проектування.

1.3 Аналіз відомих рішень в управлінні технологічними процесами

Автоматизована система управління технологічним процесом (АСУ ТП) - це цілісне рішення технічних і програмних засобів, які призначені для автоматизації управління технологічним обладнанням на промислових підприємствах. АСУ ТП забезпечує автоматизацію основних операцій технологічного процесу на об'єкті в цілому або якоїсь його частини.

АСУ ТП створюється як розподілена інтегрована система із забезпеченням конструктивної, функціональної і інтерфейсної незалежності її підсистем, що забезпечує збереження функцій окремих підсистем і їх елементів при відмовах в АСУ ТП, в тому числі на окремих рівнях її ієрархії.

АСУ ТП набули широкого розповсюдження в різних областях людської діяльності. Ними можуть бути обладнані як приватні будинки, ресторани, магазини, так і стратегічні об'єкти державної ваги, такі як аеропорти, транспортні вузли, АЕС та інші виробництва.

Розглянемо існуюче програмне забезпечення для роботи в таких системах.

CitectSCADA

CitectSCADA [4] (SCADA, supervisory control and data acquisition - диспетчерське управління і збір даних) - програмний продукт, що представляє собою повнофункціональну систему моніторингу, управління і збору даних, яка забезпечує:

- візуалізацію процесу в графічному режимі;
- «просунуте» управління алармами;
- відстеження трендів в реальному часі і доступ до архівних трендів;

- підготовку деталізованих звітів;
- статичний контроль процесу;
- багатопотокове виконання програм, розроблених на CitectVBA [5] і CiCode [7].

CitectSCADA побудована на базі мультизадачного ядра реального часу, що забезпечує продуктивність збору до 5000 значень в секунду при роботі в мережному режимі з декількома станціями. Модульна клієнт-серверна архітектура дозволяє однаково ефективно застосовувати CitectSCADA як в малих проектах з використанням тільки одного автоматизованого робочого місця (APM), так і у великих з розподілом завдань на кілька комп'ютерів.

У CitectSCADA резервування є вбудованим і мобільним. Резервування дозволяє захищати всі зони потенційних відмов як функціональних модулів (Серверів та клієнтів), так і мережових з'єднань між вузлами та пристроями введення / виведення.

CitectSCADA має вбудовану мову програмування CiCode, а також підтримку VBA (Visual Basic for Applications - спрощена реаліція мови Visual Basic). CitectSCADA працює з 32-розрядними додатками Windows 9x / NT / 2000 / XP / 2003 / Vista / 7. Збір даних, формування алармів і побудова трендів проходить одночасно з редагуванням і компіляцією.

Simplicity

Simplicity HMI [7] - це рішення по візуалізації і управління на базі архітектури клієнт-сервер, забезпечує візуалізацію операцій, автоматизований контроль і представлення надійної інформації для аналітичних додатків більш високого рівня.

Основними особливостями Simplicity є:

- підтримка до 100 серверів;
- архівування;

- відкритість архітектури дозволяє програмному забезпеченню SIMPLICITY легко взаємодіяти з іншими системами на всіх рівнях управління;
- поділ завдань на високопріоритетні і фонові;
- різноманітність власних драйверів і зв'язок з пристроями OPC (Object Linking and Embedding for Process Control - сімейство пропрограмих технологій, надає єдиний інтерфейс для управління об'єктами автоматизації і технологічними процесами);
- можливість легко масштабувати систему;
- інтеграція з ERP (Enterprise Resource Planning – планування ресурсів підприємства) і іншими зовнішніми системами;
- можливість поділу користувачів по ролям і доступним ресурсам;
- можливість відтворення заархівованих подій через засоби Digital Graphical Replay;
- сигналізація в режимі реального часу і в запису.

Master SCADA

Master SCADA [7] – програмний пакет для проектування систем диспетчерського управління та збору даних. Основними властивостями є модульність, масштабованість і об'єктний підхід до розробки.

Система призначена для збору, архівування, відображення даних, а також для управління різними технологічними процесами. Окрім створення так званого верхнього рівня, система дозволяє програмувати контролери з відкритою архітектурою. Таким чином, Master SCADA дозволяє створювати єдиний комплексний проект автоматизації (SCADA-система і ПЛК (програмований логічний контролер)). Вся система, включаючи всі комп'ютери і контролери, конфігурується в єдиному проекті, в результаті чого не треба конфігурувати внутрішні зв'язки в системі.

Основними особливостями Master SCADA є:

- єдине середовище розробки всього проекту;
- роздільне конфігурація структури системи і логічної структури об'єкта;
- відкритість і дотримання стандартам;
- необмежена гнучкість обчислювальних можливостей;
- об'єктний підхід.

TRACE MODE

TRACE MODE [8] – інтегрована інформаційна система для управління промисловим виробництвом, що об'єднує в єдине ціле продукти класу SOFTLOGIC-SCADA / HMI-MES-EAM-HRM. TRACE MODE дає рішення для управління технологічними процесами в реальному часі, здійснюваного в тісній інтеграції з керуванням виробничим бізнесом. На одній платформі об'єднані продукти для автоматизації технологічних процесів і бізнес-процесів (АСУП).

Інтегроване середовище розробки TRACE MODE це єдина програмна оболонка, що об'єднує такі основні компоненти інструментальної системи:

- SOFTLOGIC - систему програмування контролерів;
- SCADA / HMI - систему розробки розподіленої АСУ ТП;
- MES-EAM-HRM – економічні модулі для створення АСУП, повністю інтегрованої з АСУ ТП.

Основними особливостями TRACE MODE є:

- легкість інтеграції зі стороннім програмним / апаратним оточенням;
- інтеграція SCADA і Softlogic систем;
- інтегроване середовище розробки, яке об'єднує в собі більше 10 різних редакторів АСУ ТП і АСУП;
- принцип єдиного проекту для розподіленого АСУ;

- велика бібліотека драйверів для контролерів і пристроїв зв'язку з об'єктом;
- підтримка мов програмування алгоритмів міжнародного стандарту IEC 61131-3 [7];
- власний генератор звітів, що дозволяє створювати повнофункціональні HTML-звіти в реальному часі;
- технології гарячого резервування - дублювання;
- інтеграція з базами даних і другими додатками.

InTouch

Wonderware InTouch [8] це додаток - генератор графічного операторського інтерфейсу (HMI) для систем SCADA і інших систем автоматизації виробництва. InTouch дозволяє користувачам створювати операторські інтерфейси під Windows, які тісно взаємодіють з іншими.

Додаток для спільної роботи в реальному часі, планування проектів, оптимізації робочих процесів і відстеження статусу дозволять вашій команді робити більше і швидше.

Компонентами Factory Suite. Wonderware InTouch пропонує розширений набір готових Майстер-об'єктів з індустріальною графікою, які дозволяють розробнику створювати складні і потужні екрани операторського інтерфейсу.

InTouch також має потужну мову сценаріїв. Програмний пакет InTouch складається з двох основних компонентів - середовища розробки та середовища виконання. В середовищі розробки створюються мнемосхеми, визначаються і прив'язуються до апаратних засобів вхідні та вихідні сигнали і параметри, розробляються алгоритми управління і призначаються права операторів.

Збудований таким чином додаток функціонує в середовищі виконання.

Для того щоб додаток могло обмінюватись даними з апаратурою, необхідне використання третього компонента - окремої програми, званої

сервером введення / виведення. Як правило, сервер вводу / виводу орієнтований на використання з конкретним видом обладнання, таким як промислові контролери. Разом з тим використовуються також сервери введення / виведення, розраховані на обмін даними відповідно до визначених промислових стандартів і які можуть працювати з усіма контроллерами, що задовольняють цьому стандарту (наприклад, Modbus, ProfiBus, DeviceNet і ін.)

Найбільш близькою до створюваної технології є експертна система багатофакторного аналізу технологічного процесу біоконверсії реалізована в середовищі FUZZY EXPERT [6] на засадах нечіткої логіки.

Середовище FUZZY EXPERT також дозволяє користувачам створювати операторські інтерфейси під Windows, які тісно взаємодіють з іншими компонентами та користувачами.

1.4 Формулювання вимог до роботи удосконаленої технології

На основі проведених досліджень ми можемо сформулювати наступні вимоги до розроблюваної технології управління біоконверсією.

Для моделювання роботи інформаційної технології управління біоконверсією варто застосовувати апарат нечіткої логіки.

Сама технологія складається із п'яти етапів – формалізація, дерево рішень, фазифікація, база знань, налаштування (див. рис.1.1);



Рисунок 1.1 – Етапи інформаційної технології управління біоконверсією

На першому етапі описується предметна область експертної системи, визначається вихідна змінна ідентифікованого об'єкта (з безперервним виходом) та вектор вхідних змінних (з безперервним та дискретним виходом).

На другому етапі (структурної ідентифікації) формується перевернуте ієрархічне дерево виведення результату, показника виходу біогазу – нечіткої логічної змінної, значення якої відповідають нечітким лінгвістичним термам.

На вході дерева (гілках) формується вектор нечітких логічних змінних (параметрів). Кожна з цих змінних може приймати одне із значень, що відповідають нечітким логічним термам, закріпленим за інтервалами деякої універсальної шкали значень параметра.

До вектору вхідних змінних базової технології треба додати додаткові фактори впливу на процес біоконверсії, а саме: стабільність температурного режиму та якість додаткового обладнання

Прив'язка значень нечітких змінних до нечітких лінгвістичних термів називається формалізацією системи.

Наступним етапом структурної ідентифікації є фазифікація системи – постановка у відповідність кожному лінгвістичному терму функції належності. На цьому кроці визначаються моделі функцій належності, які використовуються для формалізації термів - оцінок змінних. Для цього система повинна згенерувати відповідне діалогове вікно.

Слідуючим етапом інформаційної технології є формування експертом бази знань у вигляді послідовності логічних висловлювань ЯКЩО–ТО. Ці висловлювання поєднують значення (лінгвістичні терми) вхідного вектора параметрів із певним лінгвістичним термом виходу. Ці послідовності називають продукційними правилами, а базу знань – продукційною базою знань. Експертні правила ЯКЩО-ТО вносяться до відповідних матриці знань

Для зручності роботи алгоритмів пошуку в базі знань, її спочатку представляють у вигляді нечітких логічних рівнянь, а потім у вигляді міні-максних рівнянь (дефазифікація бази знань).

В процесі налаштування (навчання) коректуються параметри функцій належності. Налаштування нечіткої експертної системи шляхом вирішення завдань оптимізації з використанням навчальної вибірки.

В процесі аналізу удосконалена модель управління повинна аналізувати наступні параметри:

- якість сировини;
- вологість сировини;
- гранулометричний склад;
- кількість жирних кислот;
- водневий показник;
- час зброджування;
- температура бродіння;
- якість перемішування;
- вміст метаногенної мікрофлори;
- доза завантаження реактора;
- тип біореактора;
- ступінь іммобілізації;
- якість додаткового обладнання;
- якість апаратно-схемного оформлення.

1.5 Висновок

В першому розділі магістерської кваліфікаційної роботи детально висвітлені проблеми, пов'язані з управлінням біоконверсією. Розглянуто технологічні фактори, що впливають на управління біоконверсією.

Розглянуто, проаналізовано та вибрано методологію управління біоконверсією. Зроблено аналіз відомих рішень для задачі управління біоконверсією. Розглянута можливість використання програмної оболонки FUZZY EXPERT, призначеної для проектування і налаштування експертних

систем на базі нечіткої логіки. Детально розглянуті та проаналізовані етапи існуючої інформаційної технології на основі програмної оболонки FUZZY EXPERT.

Прийнято рішення про удосконалення базової інформаційної технології за рахунок додавання двох додаткових факторів впливу на процес біоконверсії, а саме: стабільність температурного режиму та якість додаткового обладнання.

2. МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ УПРАВЛІННЯ БІОКОНВЕРСІЄЮ З ВИКОРИСТАННЯМ НЕЧІТКОЇ ЛОГІКИ

2.1 Обґрунтування вибору методології для моделювання роботи інформаційної технології.

В задачах прийняття рішень одним із центральних складових є поняття ідентифікації об'єкта – це побудова його математичної моделі, що встановлює взаємозв'язок між вхідними і вихідними змінними по експериментальним даним. Задача ідентифікації, як правило, вирішується в два етапи.

На першому етапі, що називається структурною ідентифікацією, формується груба модель об'єкта, яка апроксимує зв'язок вхід-вихід і містить параметри налаштування.

Етап структурної ідентифікації є швидше мистецтвом, чим наукою, оскільки вибір початкової моделі залежить від «інтелектуального капіталу дослідника»: його освіти, досвіду, розуміння сутності об'єкта, міри уподобання до того, чи іншого математичного апарату та інших суб'єктивних факторів. В якості налаштовуваних моделей в сучасній теорії ідентифікації використовуються кількісні співвідношення у вигляді рівнянь: алгебраїчних, диференціальних, різностних, інтегральних т. і. Цей апарат природньо застосовується до тих об'єктів, що описуються законами фізики: механіки, термодинаміки, електромагнетизма. В той же час класична теорія породжує катастрофічно складні моделі при спробі ідентифікувати залежності в так званих інтелектуальних задачах, які традиційно вирішуються людьми. Людина ходить, плаває, виконує складні гімнастичні вправи, водить автомобіль, пізнає знайомі об'єкти, знаходить закономірності в експериментальних даних, вирішує інші, надзвичайно складні з математичної точки зору задачі управління та прийняття рішень, не використовуючи строгих математичних співвідношень.

Фундаментальну роль у вирішенні людиною задач ідентифікації та прийняття рішень грають дві унікальні властивості:

навчання, - як здатність послідовно мінімізувати відхилення фактичного результату від деякого бажаного еталону;

лінгвістичність – як здатність висловлювати природною мовою ті знання, що одержані в результаті навчання.

Тому, моделюючи інтелектуальну діяльність, природно звертатись до такого математичного апарату, який на відміну від класичних методів, пристосований до врахування здатності навчатись та лінгвістичності.

Всі сучасні інформаційні інтелектуальні технології які застосовуються для вирішення задач ідентифікації та прийняття рішень являють собою комплексне застосування трьох, незалежних одна від одної теорій:

- нечітких множин – засіб формалізації природно-мовних висловлювань то логічного виводу;
- нейронних мереж – штучні аналоги людського мозку, що мають здатність до навчання;
- генетичних алгоритмів – методу синтеза оптимальних рішень із множини початкових варіантів, над якими виконуються операції схрещування, мутації та селекції.

Для вибору методу моделювання інформаційної технології управління біоконверсією розглянемо більш детально три найпоширеніші методології інтелектуальної ідентифікації.

2.1.1 Нечіткі множини

Логіка в звичайному сенсі слова є уявленням механізму мислення, повинна бути завжди строго формалізованою. Однак в дійсності існує не одна логіка (наприклад, булева), а стільки, скільки ми побажаємо, тому що все визначається вибором відповідної системи аксіом. Звичайно, як тільки аксіоми прийняті, всі твердження, що побудовані на їх основі, повинні бути строгими,

без протиріч пов'язані за правилами, встановленими в цій системі аксіом.

Нечітка логіка є узагальненням класичної логіки на випадок, коли істинність розглядається як лінгвістична змінна, що приймає значення типу: "дуже істинно", "більш-менш істинно", "не дуже хибно" і т.п.

Зазначені лінгвістичні значення представляються нечіткими множинами. Основна відмінність від класичної логіки полягає в тому, що замість значень "Істина" і "Хибність" в нечіткій логіці використовується ступінь істинності, що приймає значення з нескінченної множини від 0 (Хибність) до 1 (Істина) включно.

Нечіткі множини – це засоби формалізації природно-лінгвістичних висловлювань та логічних висновків. Ідея, що лежить в основі формалізації причинно-наслідкових зв'язків між змінними «входи-виходи», полягає в описі цих зв'язків на природній мові з використанням теорії нечітких множин та лінгвістичних змінних. Моделі об'єктів будуються шляхом проектування та налаштування нечітких баз знань, що представляють собою сукупності лінгвістичних висловлювань типу ЯКЩО <входи>, ТО <виходи>.

Ідея, що лежить в основі формалізації причинно-наслідкових зв'язків між змінними «входи-виходи», полягає в описі цих зв'язків на природній мові з використанням теорії нечітких множин та лінгвістичних змінних. Ціль даного підрозділу полягає у введенні основних формалізмів, необхідних для визначення нечітких баз знань, що є носієм експертної інформації. В основу цього підрозділу покладена робота [6].

Входи і виходи об'єкта

Нами розглядається об'єкт з одним виходом та n входами виду:

$$y = f_y(x_1, x_2, \dots, x_n) \quad , \quad (2.1)$$

де y – вхідна змінна; x_1, x_2, \dots, x_n – вихідні змінні.

Змінні x_1, x_2, \dots, x_n та y можуть бути кількісними і якісними.

Прикладами кількісних змінних є:

ТЕМПЕРАТУРА РЕАКТОРА = [30, 41] °С, ДОЗА ЗАВАНТАЖЕННЯ РЕАКТОРА = [6, 20]%, та інші змінні, які легко вимірюються в прийнятих для них кількісних шкалах.

Прикладом змінної, для якої не існує природної кількісної шкали, є РІВЕНЬ СТРЕСУ ОПЕРАТОРА, який може бути оцінений якісними термами (низький, середній, високий) або вимірюватися в штучних шкалах, наприклад, по 5-бальній, 10-бальній, ..., 100-бальній системах.

Для кількісних змінних передбачаються відомими зміни:

$$U_i = [\underline{x}_i, \bar{x}_i], i = \overline{1, n}, \quad (2.2)$$

$$Y = [\underline{y}, \bar{y}] , \quad (2.3)$$

де \underline{x}_i (\bar{x}_i) - нижнє (верхнє) значення вхідної змінних \bar{x}_j , $i = \overline{1, n}$,

\underline{y} (\bar{y}) – нижнє (верхнє) значення вихідної змінної y .

Для якісних змінних $x_1 \div x_n$ та y передбачається, що змінюються множини всіх можливих значень:

$$U_i = [v_i^1, v_i^2, \dots, v_i^{q_i}], i = \overline{1, n} , \quad (2.4)$$

$$Y = [y^1, y^2, \dots, y^{q_m}] , \quad (2.5)$$

де v_i^l ($v_i^{q_i}$) - бальна оцінка, що відповідає найменшому (найбільшому) значенню вхідної змінної x_j ;

y^1 (y^{q_m}) - бальна оцінка, що відповідає найменшому (найбільшому) значенню вихідної змінної y ;

$q_j, i = \overline{1, n}$ та q_m - потужності множин (3.4) та (3.5), при чому в загальному випадку $q_1 \neq q_2 \neq \dots \neq q_n \neq q_m$.

Лінгвістичні змінні

Нехай $X^* = \langle x_1^*, x_2^*, \dots, x_n^* \rangle$ - вектор фіксованих значень вхідних змінних розглянутого об'єкта, де $x_i^* \in U_i, i = \overline{1, n}$. Задача прийняття рішення полягає у тому, щоб на основі інформації про вектор входів X^* визначити вихід $y^* \in Y$. Необхідною умовою формального рішення такої задачі є наявність залежності (3.1). Для встановлення цієї залежності будемо розглядати вхідні змінні $x_j, i = \overline{1, n}$ та вхідну змінну y як лінгвістичні змінні [15], задані на універсальних множинах (2.2), (2.3) або (2.4), (2.5).

Для оцінки лінгвістичних змінних $x_j, i = \overline{1, n}$ та y будемо використовувати якісні терми з наступних терм-множин:

$A_i = \{a_i^1, a_i^2, \dots, a_i^{l_i}\}$ – терм-множина змінної $x_j, i = \overline{1, n}$,

$D = \{d_1, d_2, \dots, d_m\}$ - терм-множина y ,

де a_i^p - p -й лінгвістичний терм змінної $x_j, p = \overline{1, l_j}, i = \overline{1, n}$.

d_j - j -й лінгвістичний терм змінної y ,

m – число різноманітних рішень області, що розглядається.

Потужності терм-множин $A_i, i = \overline{1, n}$ в загальному випадку можуть бути відмінними, тобто $l_1 \neq l_2 \neq \dots \neq l_n$.

Назви окремих термов $a_i^1, a_i^2, \dots, a_i^{l_i}$ можуть також відрізнятися один від одного для різноманітних лінгвістичних змінних $x_j, i = \overline{1, n}$. Наприклад, ТЕМПЕРАТУРА КОНВЕРСІЇ {психрофільна, мезофільна, термофільна}.

Лінгвістичні терми $a_i^P \in A_i$ та $d_j \in D$, $p = \overline{1, l_j}$, $i = \overline{1, n}$, $j = \overline{1, m}$ будемо розглядати як нечіткі множини, задані на універсальних множинах U_i та Y , визначинних співвідношеннями (2.2)÷(2.5).

В випадку кількісних змінних $x_j, i = \overline{1, n}$ та y , нечіткі множини a_i^P та d_j , визначені співвідношеннями:

$$a_i^P = \int_{\underline{x_i}}^{\overline{x_i}} \mu^{a_i^P}(x_i) / x_i, \quad (2.6)$$

$$d_j = \int_{\underline{d}}^{\overline{d}} \mu^{d_j}(d) / d, \quad (2.7)$$

де, $\mu^{a_i^P}(x_i)$ - функція приналежності значення вхідної змінної $x_i \in [\underline{x_i}, \overline{x_i}]$ терму $a_i^P \in A_i$, $p = \overline{1, l_j}$, $i = \overline{1, n}$;

$\mu^{d_j}(d)$ – функція приналежності значення вихідної змінних $y \in [\underline{y}, \overline{y}]$ терму-рішенню $d_j \in D$, $j = \overline{1, m}$.

В випадку якісних змінних x_j , $i = \overline{1, n}$ та y нечіткі множини a_i^P та d_j визначимо так:

$$a_i^P = \sum_{k=1}^{q_i} \mu^{a_i^P}(v_i^k) / v_i^k, \quad (2.8)$$

$$d_j = \sum_{r=1}^{q_m} \mu^{d_j}(y^r) / y^r, \quad (2.9)$$

де $\mu^{a_i^P}(v_i^k)$ - ступінь приналежності елемента $v_i^k \in U_i$, терму $a_i^P \in A_i$, $p = \overline{1, l_j}$, $i = \overline{1, n}$, $k = \overline{1, q_j}$;

$\mu^{d_j}(y^r)$ – ступінь приналежності елемента $y^r \in Y$ терму-рішенню $d_j \in D$, $j = \overline{1, m}$.

U_i та Y визначається співвідношеннями (2.4) та (2.5).

Зауважимо, що в співвідношеннях (2.6)-(2.9) знаки інтегралу та суми позначають об'єднання пар $\mu(u)/u$.

Даний етап побудови нечіткої моделі, на якому визначаються лінгвістичні оцінки змінних та необхідних для їх формалізації функції приналежності, отримав в літературі по нечіткій логіці [8] назву *фаззифікації змінних* (від англ. fuzzification).

Нечітка база знань

Візьмемо N експериментальних даних, які зв'язують входи і виходи об'єкта ідентифікації, і розподілимо їх наступним чином:

$$N = k_1 + k_2 + \dots + k_m,$$

Де k_j – число експериментальних даних, які відповідають вихідному рішенню $d_j, j = \overline{1, m}$, m – число вихідних рішень, причому, в загальному випадку $k_1 \neq k_2 \neq \dots \neq k_m$.

Передбачається, що $N < l_1 \cdot l_2 \cdot \dots \cdot l_n$, тобто число відібраних експериментальних даних менше повного перебору різних поєднань рівнів ($l_i, i = \overline{1, n}$) зміни вхідних змінних об'єкта.

Пронумеруємо N експериментальних даних наступним чином:

11, 12, $1k_1$ – номери комбінацій вхідних змінних для вирішення d_1 ;

21, 22, $2k_2$ – номери комбінацій вхідних змінних для вирішення d_2 ;

...

j_1, j_2, jk_j – номери комбінацій вхідних змінних для вирішення d_j ;

...

m_1, m_2, \dots, m_k – номери комбінацій вхідних змінних для вирішення d_m .

Матрицею знань [34] назвемо таблицю, сформовану за такими правилами (див табл.2.1):

- 1) Розмірність матриці дорівнює $(n + 1) \times N$, де $(n + 1)$ – число стовбців, а $N = k_1 + k_2 + \dots + k_m$ – число рядків.
- 2) Перші n стовбців матриці відповідають вхідним змінним x_i , $i = \overline{1, n}$, а $(n + 1)$ – й стовбець відповідає значенням d_j вихідної змінної y ($j = \overline{1, m}$).
- 3) Кожен рядок матриці являє собою деяку комбінацію значень вхідних змінних, віднесених експертом до одного з можливих значень вихідної змінної y . При цьому : перші k_1 рядків відповідають значенню вихідної змінної $y = d_1$, другі k_2 рядків $y = d_2, \dots$, останні k_m рядків – значенню $y = d_m$
- 4) Елемент a_i^{jp} , який стоїть на перетині i -го стовбця і jp -го рядка відповідають лінгвістичній оцінці параметра x_i в рядку нечіткої бази даних з номером jp . При цьому лінгвістична оцінка a_i^{jp} вибирається із терм-множини, яка відповідає змінній x_i , тобто $a_i^{jp} \in A_i$, $i = \overline{1, n}$, $j = \overline{1, m}$, $p = \overline{1, k_j}$

Таблиця 2.1 - Матриця знань

Номер вхідної комбінації значень	Вхідні змінні				Вихідна змінна
	x_1	x_2	$\dots x_i \dots$	x_n	
11	a_1^{11}	a_2^{11}	$\dots a_i^{11} \dots$	a_n^{11}	d_1
12	a_1^{12}	a_2^{12}	$\dots a_i^{12} \dots$	a_n^{12}	
...					
1k ₁	$a_1^{1k_1}$	$a_2^{1k_1}$	$\dots a_i^{1k_1} \dots$	$a_n^{1k_1}$	
...					
j1	a_1^{j1}	a_2^{j1}	$\dots a_i^{j1} \dots$	a_n^{j1}	d_j
j2	a_1^{j2}	a_2^{j2}	$\dots a_i^{j2} \dots$	a_n^{j2}	
...					
jk _j	$a_1^{jk_j}$	$a_2^{jk_j}$	$\dots a_i^{jk_j} \dots$	$a_n^{jk_j}$	
...					
m1	a_1^{m1}	a_2^{m1}	$\dots a_i^{m1} \dots$	a_n^{m1}	d_m
m2	a_1^{m2}	a_2^{m2}	$\dots a_i^{m2} \dots$	a_n^{m2}	
...					
mk _m	$a_1^{mk_m}$	$a_2^{mk_m}$	$\dots a_i^{mk_m} \dots$	$a_n^{mk_m}$	

Введена матриця знань визначає систему логічних висловлювань типу «ЯКЩО-ТО, ІНАКШЕ», які пов'язують значення вхідних змінних $x_1 \div x_n$ з одним з можливих типів вирішення $d_j, j = \overline{1, m}$:

ЯКЩО $(x_1 = a_1^{11}) \text{ I } (x_2 = a_2^{11}) \text{ I } \dots \text{ I } (x_n = a_n^{11})$ АБО
 $(x_1 = a_1^{12}) \text{ I } (x_2 = a_2^{12}) \text{ I } \dots \text{ I } (x_n = a_n^{12})$ АБО...
 $(x_1 = a_1^{1k_1}) \text{ I } (x_2 = a_2^{1k_1}) \text{ I } \dots \text{ I } (x_n = a_n^{1k_1}),$

ТО $y = d_1$, ІНАКШЕ

ЯКЩО $(x_1 = a_1^{21}) \text{ I } (x_2 = a_2^{21}) \text{ I } \dots \text{ I } (x_n = a_n^{21})$ АБО
 $(x_1 = a_1^{22}) \text{ I } (x_2 = a_2^{22}) \text{ I } \dots \text{ I } (x_n = a_n^{22})$ АБО...
 $(x_1 = a_1^{2k_2}) \text{ I } (x_2 = a_2^{2k_2}) \text{ I } \dots \text{ I } (x_n = a_n^{2k_2}),$

ТО $y = d_2$, ІНАКШЕ

ЯКЩО $(x_1 = a_1^{m1}) \text{ I } (x_2 = a_2^{m1}) \text{ I } \dots \text{ I } (x_n = a_n^{m1})$ АБО
 $(x_1 = a_1^{m2}) \text{ I } (x_2 = a_2^{m2}) \text{ I } \dots \text{ I } (x_n = a_n^{m2})$ АБО...
 $(x_1 = a_1^{mk_m}) \text{ I } (x_2 = a_2^{mk_m}) \text{ I } \dots \text{ I } (x_n = a_n^{mk_m}),$

ТО $y = d_m$,

де $d_j (j = \overline{1, m})$ – лінгвістична оцінка вихідної змінної y , яка визначається із терм множини D ;

a_i^{jp} – лінгвістична оцінка змінної x_i в p -ому рядку j -ї диз'юнкції, яка вибирається з терм множини $A_i, i = \overline{1, n}, j = \overline{1, m}, p = \overline{1, k_j}$;

k_j – кількість правил, які визначають значення вихідної змінної $y = d_j$.

Будемо називати подібну систему логічних висловлювань *нечіткою базою знань*.

З використанням операцій \cup (АБО) і \cap (І) система логічних висловлювань (3.10) може бути переписана у більш компактному вигляді:

$$\cup_{p=1}^{kj} [\cap_{i=1}^n (x_i = a_i^{jp})] \rightarrow y = d_j, j = \overline{1, m}. \quad (2.11)$$

Таким чином шукане співвідношення (3.1), яке встановлює зв'язок між вхідними параметрами x_i і вихідною змінною y , формалізовано у вигляді системи нечітких логічних висловлювань (3.11), яка базується на введеній нами матриці знань (табл. 3.1).

Функції належності

За визначенням [6], функція належності $\mu^T(x)$ характеризує суб'єктивну міру (в діапазоні $[0,1]$) впевненості експерта в тому, що чітке значення x відповідає нечіткому терму T . Найбільше розповсюдження в практичних додатках отримали трикутні, трапецієподібні, дзвоноподібні (гаусові) функції належності, параметри яких дозволяють міняти форму функцій.

Нами пропонується проста і зручна для налаштування аналітична модель функцій належності змінної x довільному нечіткому терму T у вигляді:

$$\mu^T(x) = \frac{1}{1 + \left(\frac{x-b}{c}\right)^2},$$

Де b і c – параметри налаштування; b – координата максимуму функції $\mu^T(b) = 1$; c – коефіцієнт концентрації – розтягу функції

Для нечіткого терма T число b являє собою найбільш можливе значення змінної x .

Постановка задачі апроксимації

Будемо вважати відомими:

- множину рішень $D = \{d_1, d_2, \dots, d_m\}$, які відповідають вихідній змінній y ,
- множину вхідних змінних $X = (x_1, x_2, \dots, x_n)$,
- діапазони множинного вимірювання кожної вхідної змінної $x_i \in [\underline{x}_i, \overline{x}_i], i = \overline{1, n}$
- функції належностей, які дозволяють представляти змінні $x_i, i = \overline{1, n}$ у вигляді нечітких множин (3.6) або (3.8),
- матриця знань, визначена за правилами, введеними у розділі 3.1.3

Потрібно: розробити алгоритм прийняття рішення, який дозволяє фіксованому вектору вхідних змінних $X^* = \langle x_1^*, x_2^*, \dots, x_n^* \rangle, x_i^* \in [\underline{x}_i, \overline{x}_i]$ поставити у відповідність розв'язок $y \in D$.

Ідея методу, запропонованого нижче для розв'язання цієї задачі полягає в використанні нечітких логічних рівнянь. Ці рівняння будуються на базі матриці знань чи ізоморфної їй системі логічних висловлювань (2.10) і дозволяють обчислювати значення функцій належності різних розв'язків при фіксованих значеннях вхідних змінних об'єкта. В якості шуканого розв'язку вибирається розв'язок з найбільшим значенням функції належності.

Нечіткі логічні рівняння

Лінгвістична оцінка a_i^{jp} змінних $x_1 \div x_n$, які входять в логічне висловлювання про розв'язки d_j (3.10), будемо розглядати як нечіткі

множини, визначені на універсальних множинах $U_i = [\underline{x}_i, \overline{x}_i]$, $i = \overline{1, n}$, $j = \overline{1, m}$.

Нехай $\mu^{a_i^{jp}}(x_i)$ – функція належності параметра $x_i \in [\underline{x}_i, \overline{x}_i]$ нечіткому терму a_i^{jp} , $i = \overline{1, n}$, $j = \overline{1, m}$, $p = \overline{1, k_j}$;

$\mu^{d_j} = (x_1, x_2, \dots, x_n)$ – залежна від n змінних функція належності вектора вхідних змінних $X = (x_1, x_2, \dots, x_n)$ значенню вихідної змінної $y = d_j$, $j = \overline{1, m}$.

Зв'язок між цими функціями визначається нечіткою базою знань (2.11) і може бути представлена у вигляді наступних рівнянь

$$\begin{aligned} \mu^{d_1}(x_1, x_2, \dots, x_n) = & \mu^{a_1^{11}}(x_1) \wedge \mu^{a_2^{11}}(x_2) \wedge \dots \wedge \mu^{a_n^{11}}(x_n) \vee \\ & \vee \mu^{a_1^{12}}(x_1) \wedge \mu^{a_2^{12}}(x_2) \wedge \dots \wedge \mu^{a_n^{12}}(x_n) \vee \dots \\ & \dots \vee \mu^{a_1^{1k_1}}(x_1) \wedge \mu^{a_2^{1k_1}}(x_2) \wedge \dots \wedge \mu^{a_n^{1k_1}}(x_n), \end{aligned}$$

$$\begin{aligned} \mu^{d_2}(x_1, x_2, \dots, x_n) = & \mu^{a_1^{21}}(x_1) \wedge \mu^{a_2^{21}}(x_2) \wedge \dots \wedge \mu^{a_n^{21}}(x_n) \vee \\ & \vee \mu^{a_1^{22}}(x_1) \wedge \mu^{a_2^{22}}(x_2) \wedge \dots \wedge \mu^{a_n^{22}}(x_n) \vee \dots \\ & \dots \vee \mu^{a_1^{2k_2}}(x_1) \wedge \mu^{a_2^{2k_2}}(x_2) \wedge \dots \wedge \mu^{a_n^{2k_2}}(x_n), \end{aligned}$$

...

$$\begin{aligned} \mu^{d_m}(x_1, x_2, \dots, x_n) = & \mu^{a_1^{m1}}(x_1) \wedge \mu^{a_2^{m1}}(x_2) \wedge \dots \wedge \mu^{a_n^{m1}}(x_n) \vee \\ & \vee \mu^{a_1^{m2}}(x_1) \wedge \mu^{a_2^{m2}}(x_2) \wedge \dots \wedge \mu^{a_n^{m2}}(x_n) \vee \dots \\ & \dots \vee \mu^{a_1^{mk_m}}(x_1) \wedge \mu^{a_2^{mk_m}}(x_2) \wedge \dots \wedge \mu^{a_n^{mk_m}}(x_n), \end{aligned}$$

Де \vee - логічне АБО, \wedge - логічне І,

Ці нечіткі логічні рівняння отримані з нечіткої бази знань (2.11) шляхом заміни лінгвістичних термів a_i^{jp} і d_j на відповідні функції належності, а операції \cup та \cap - на операції \vee і \wedge .

Коротко систему рівнянь можна записати наступним чином:

$$\mu^{d_j}(x_1, x_2, \dots, x_n) = \bigvee_{p=1}^{k_j} \left[\bigwedge_{i=1}^n \mu^{a_i^{jp}}(x_i) \right], j = \overline{1, m}. \quad (2.13)$$

Алгоритм апроксимації

Прийняття рішення $d^* \in D = \{d_1, d_2, \dots, d_m\}$, яке відповідає вектору фіксованих значень змінних $X^* = \langle x_1^*, x_2^*, \dots, x_n^* \rangle$, будемо здійснювати в такій послідовності:

1. Зафіксуємо вектор значень вхідних змінних

$$X^* = (x_1^*, x_2^*, \dots, x_n^*)$$

2. Задамо функції належності нечітких термів, які використовуються у нечіткій базі знань (3.11) і визначимо значення цих функцій для заданих значень вхідних змінних $x_1^* \div x_2^*$

3. Використовуючи логічні рівняння (3.13) обчислимо багатомірні функції належності $\mu^{d_j}(x_1^*, x_2^*, \dots, x_n^*)$ вектора X^* для всіх значень $d_j, j = \overline{1, m}$ вихідної змінної y . При цьому логічні операції \cap та \cup над функціями належності замінюються на операції \min та \max .

$$\mu(a) \wedge \mu(b) = \min[\mu(a), \mu(b)],$$

$$\mu(a) \vee \mu(b) = \max[\mu(a), \mu(b)].$$

4. Визначимо значення d_j^* , функція належності якого максимальна:

$$\mu^{d_j^*}(x_1^*, x_2^*, \dots, x_n^*) = \max_{j = \overline{1, m}} (\mu^{d_j}(x_1^*, x_2^*, \dots, x_n^*))$$

Це і буде шуканим розв'язком для вектора значень вхідних змінних $X^* = (x_1^*, x_2^*, \dots, x_n^*)$.

Таким чином, запропонований алгоритм використовує ідею ідентифікації лінгвістичного терма за максимумом функції належності, і узагальнює цю ідею на всю базу знань.

Обчислювальна частина запропонованого алгоритму легко реалізується на матриці значень функцій належності, отриманій із матриці знань шляхом виконання операцій \min та \max .

Наведений алгоритм знаходження дискретних значень $\{d_1, d_2, \dots, d_m\}$ вихідної змінної y за заданим вектором фіксованих значень вхідних змінних $X^* = \langle x_1^*, x_2^*, \dots, x_n^* \rangle$ і матриці значень дозволяє апроксимувати об'єкт $y = f_y(x_1, x_2, \dots, x_n)$ з дискретним виходом.

Оскільки лінгвістичні терми, якими оперують при створенні проектуємої системи, не мають чітко визначеного числового еквівалента, то найбільш зручною для моделювання роботи системи є теорія нечітких множин.

Моделі об'єктів в ній будуються за допомогою проектування та налаштування нечітких баз знань, які являють собою сукупності лінгвістичних висловлювань, типу ЯКЩО <вхід>, ТО <вихід>. Головна ідея полягає у тому, що налаштовуючи нечітку базу знань, можна ідентифікувати нелінійні залежності з необхідною точністю [6].

2.1.2 Нейронні мережі

Цей підрозділ написаний за матеріалами робіт [8,12]. З додатковими відомостями по нейронних мереж можна познайомитися в роботах [4,19,].

Основні поняття

Проблема машинної імітації людських думок надихає вчених вже кілька століть. Більше 50 років тому були створені перші електронні моделі нервових клітин. Крім того, з'являлися багато робіт по новим математичним моделям і навчальним алгоритмам. Сьогодні так звані нейронні мережі становлять найбільший інтерес в цій галузі. Вони використовують безліч простих обчислювальних елементів, званих нейронами, кожен з яких імітує поведінку окремої клітини людського мозку. Прийнято вважати, що людський мозок - це природна нейронна мережа, а модель мозку - це просто нейронна мережа. На рис. 2.1 показана базова структура такої нейронної мережі.

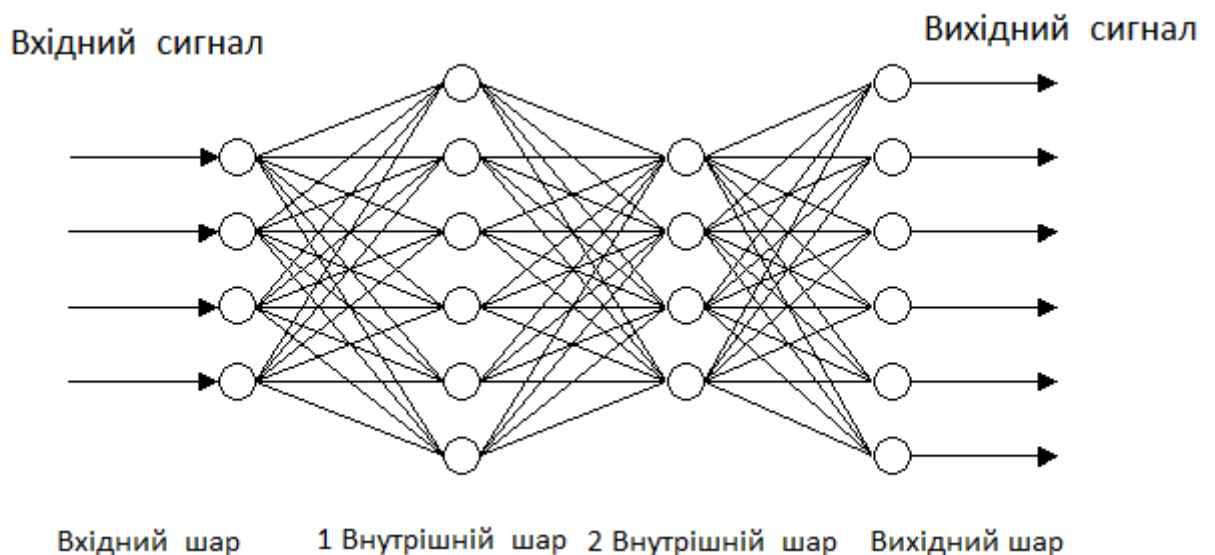


Рис. 2.1 - Базова структура нейронної мережі

Кожен нейрон в нейронній мережі здійснює перетворення вхідних сигналів у вихідний сигнал і пов'язаний з іншими нейронами. Вхідні нейрони формують так званий інтерфейс нейронної мережі. Нейронна мережа, показана на рис. 2.1, має шар, який приймає вхідні сигнали, і шар, що генерує вихідні сигнали. Інформація вводиться в нейронну мережу через вхідний шар. Всі шари нейронної мережі обробляють ці сигнали до тих пір, поки вони не досягнуть вихідного шару.

Завдання нейронної мережі - перетворення інформації потрібним чином. Для цього мережа попередньо навчається. При навчанні використовуються ідеальні (еталонні) значення пар <входи-виходи> або <учитель>, який оцінює поведінку нейронної мережі. Для навчання використовується так званий навчальний алгоритм. Ненастроєного нейронна мережа не здатна відобразити бажаної поведінки. Навчальний алгоритм модифікує окремі нейрони мережі і ваги її зв'язків таким чином, щоб поведінка мережі відповідало бажаного поведінки.

Імітація нервових клітин

Дослідники в області нейронних мереж проаналізували безліч моделей клітин людського мозку. Далі будуть розглянуті лише ті з них, які найбільш часто використовуються в промислових застосуваннях.

Людський мозок складається з більш ніж 1011 нервових клітин, що мають більш 1014 взаємозв'язків. На рис. 2.2 показана спрощена схема такого людського нейрона. Сама по собі клітина складається з ядра і зовнішньої електромембрани. Кожен нейрон має рівень активації, що лежить в діапазоні між максимумом і мінімумом, отже, на відміну від булевої логіки, існує більш ніж два рівня активації.

Для збільшення або зменшення активності даного нейрона іншими нейронами існують так звані синапси. Вони переносять величину активності від нейрона-відправника до нейрона-одержувачу. Якщо синапс є збудливим,

то величина активності нейрона-відправника збільшує активність нейрона-одержувача. Якщо синапс є гальмівним, то величина активності нейрона-відправника зменшує активність нейрона-одержувача. Синапси розрізняються не тільки за ознакою гальмування або збудження нейрона-одержувача, але також і за сумарним впливу (синаптична потужність). Вихідний сигнал кожного нейрона передається по так званому аксону, який закінчується більш ніж 10000 синапсами, що впливають на інші нейрони.

Розглянута модель нейрона лежить в основі більшості сьогоденних застосувань нейронної мережі. Відзначимо, що дана модель є лише дуже грубим наближенням дійсності. Насправді ми не можемо змоделювати навіть один єдиний людський нейрон; це вище людських можливостей у моделюванні. Отже, будь-яка робота, що базується на цій простій моделі нейрона, не спроможна точно імітувати людський мозок. Однак багато успішні застосування, що використовують цей метод, забезпечили успіх нейронних мереж, що базуються на простій моделі нейрона.

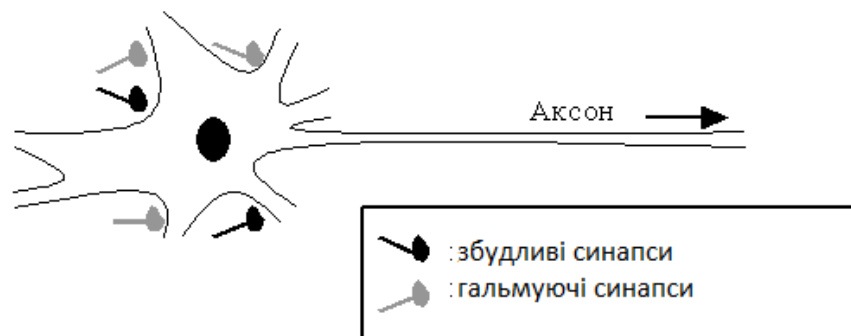


Рис. 2.2 - Спрощена схема людського нейрона.

Математична модель нейрона

Безліч математичних моделей нейрона може бути побудовано на базі простої концепції будови нейрона. На рис. 2.3 показана найбільш загальна схема. Так звана сумуюча функція об'єднує всі вхідні сигнали X_i , які надходять

від нейронів-відправників. Значним такого об'єднання є зважена сума, де ваги W_i являють собою синаптичні потужності. Збудливі синапси мають позитивні ваги, а гальмують синапси - негативні ваги. Для вираження нижнього рівня активації нейрона до зваженої суми додається компенсація (зміщення-ня) Q .

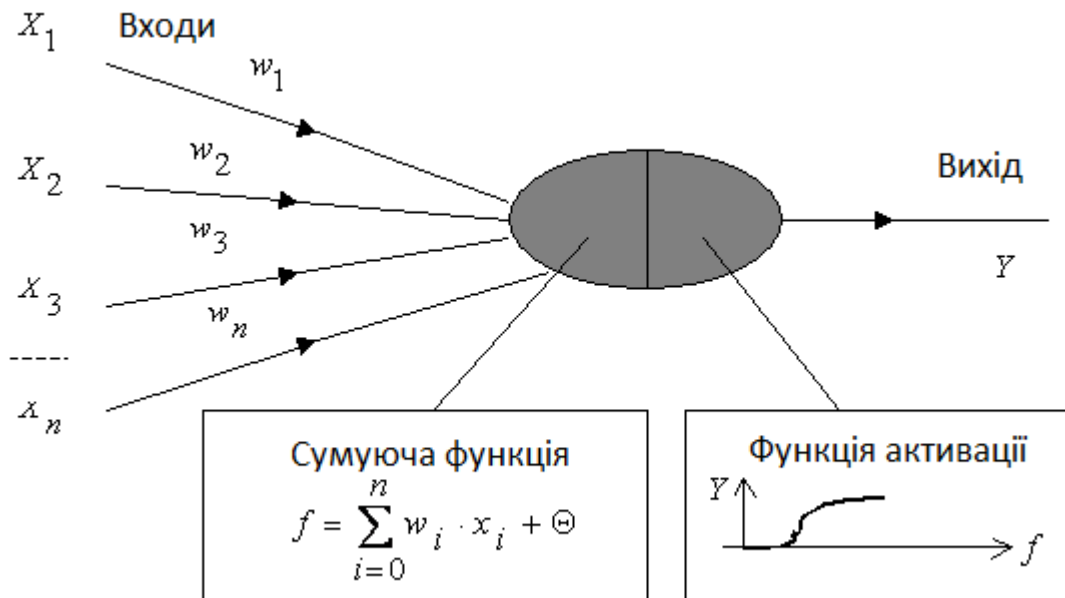


Рис. 2.3 - Проста математична модель нейрона.

Так звана функція активації розраховує вихідний сигнал нейрона Y за рівнем активності f . Функція активації зазвичай є сігмоїдною, як показано в правій нижній рамці на рис.2.4,а. Іншими можливими видами функцій активації є лінійна і радіально-симетрична функції, показані на рис.2.4,б.

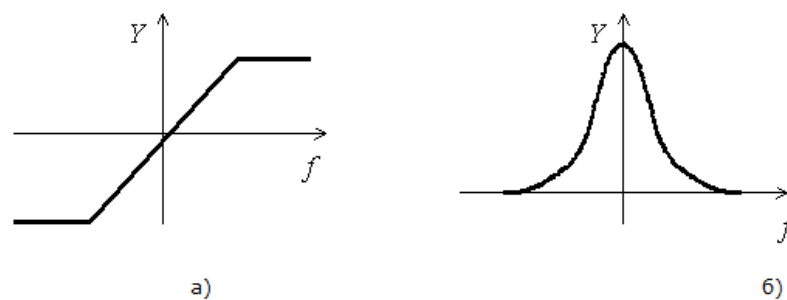


Рис. 2.4. Функції активації нейронів: (а) лінійна, (б) радіально-симетрична

Навчання нейронних мереж

Існує безліч способів побудови нейронних мереж. Вони розрізняються своєю архітектурою і методами навчання.

Перший крок у проектуванні нейронної мережі полягає в її навчанні бажаного поведінки. Це - фаза навчання. Для цього використовується так звана навчальна вибірка або вчитель. Вчитель - це або математична функція, або особа, яка оцінює якість поведінки нейронної мережі. Оскільки нейронні мережі в основному використовуються в складних застосуваннях, де немає хороших математичних моделей, то навчання проводиться за допомогою навчальної вибірки, тобто еталонних пар <входи-виходи>.

Після завершення навчання нейронна мережа готова до використання. Це - робоча фаза. В результаті навчання нейронна мережа буде обчислювати вихідні сигнали, близькі до еталонних даними при відповідних вхідних сигналах. При проміжних вхідних сигналах мережу апроксимують необхідні вихідні величини. Поведінка нейронної мережі в робочій фазі детерміновано, тобто для кожної комбінації вхідних сигналів на виході завжди будуть одні й ті ж сигнали. На протяз робочої фази нейронна мережа не навчається. Це дуже важливо для більшості технічних застосувань, оскільки система не буде прагнути до екстремального поведінки.

Собаки Павлова. Як же навчається нейронна мережа? Як правило, це демонструється на прикладі відомих собак Павлова. Коли він показував собакам їжу, у них виділялася слина. У собачих клітинах встановлювалися дзвіночки. Коли дзвонив дзвіночок, у собак не виділялася слина, т. Е. Вони не бачили зв'язку між дзвінком і їжею. Тоді Павлов став навчати собак інакше, кожен раз використовуючи дзвіночок при пред'явленні їжі. Після цього, навіть за відсутності їжі, наявність дзвіночка викликало у собак слину.

На рис. 2.5 показано, як проста модель нейрона може бути представлена на прикладі собаки Павлова. Є два вхідних нейрона: один з них відповідає тому, що собака бачить їжу, інший - наявності дзвінка. Обидва вхідних

нейрона мають зв'язку з вихідним нейроном. Ці зв'язки відповідають синапсах, а товщина ліній - вага синапсів. Перед навчанням собака реагує лише на їжу, але не на дзвінок. Отже, лінія між лівим вхідним і вихідним нейронами є жирної, в той час як лінія між правим вхідним і вихідним нейронами є дуже тонкою.

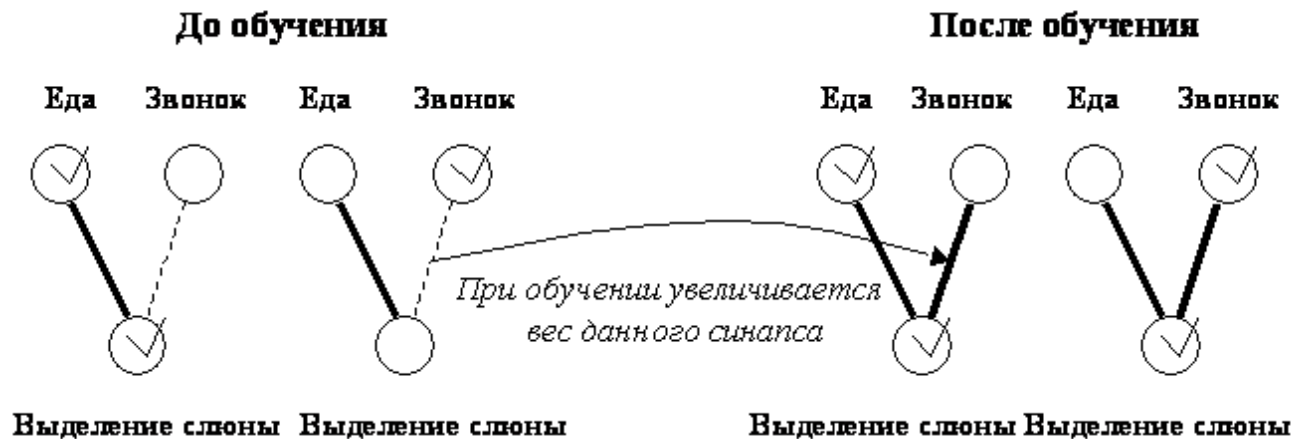


Рис. 2.5 - Принцип експерименту Павлова над собаками.

Правило навчання Хебба. Цілком очевидно, що дзвінок при пред'явленні їжі виробляє асоціацію між ним та їжею. Отже, права лінія також стає товщі, оскільки збільшується вага синапсу. На підставі цих спостережень Хебб в 1949 році запропонував наступне навчальне правило:

- збільшувати вагу активного входу нейрона, якщо вихід цього нейрона повинен бути активним.
- зменшити вагу активного входу нейрона, якщо вихід цього нейрона не повинен бути активним.

Це правило, назване правилом Хебба, передує всім навчальним правилами, включаючи найбільш використовуваний в даний час метод зворотного поширення помилки (error backpropagation algorithm).

Метод зворотного поширення помилки

Цей метод навчання багат шарової нейронної мережі називається узагальненим дельта-правилом або правилом error backpropagation (зворотного поширення помилки). Метод був запропонований в 1986 р. Румельхартом, Маккеландом і Вільямсом. Це ознаменувало відродження інтересу до нейронних мереж, який став згасати на початку 70-х років. Пізніше було виявлено, що Паркер опублікував подібні результати в 1982р, а Вербос виконав таку роботу в 1984р. Однак така природа науки, що вчені, які працюють незалежно один від одного, не можуть використовувати все те прогресивне, що є в інших областях, і тому часто трапляється повторення вже досягнутого. Однак стаття Руммельхарта та ін., Опублікована в журналі Nature (1986), є досі найбільш цитованої в цій галузі.

Навчання мережі починається з пред'явлення образу і обчислення відповідної реакції. Порівняння з бажаною реакцією дає можливість змінювати ваги зв'язків таким чином, щоб мережа на наступному кроці могла видавати більш точний результат. Навчальне правило забезпечує налаштування ваг зв'язків. Інформація про виходи мережі є вихідною для нейронів попередніх шарів. Ці нейрони можуть налаштовувати ваги своїх зв'язків для зменшення похибки на наступному кроці.

Коли ми пред'являємо неналагодженій мережі вхідний образ, вона буде видавати деякий випадковий вихід. Функція помилки являє собою різницю між поточним виходом мережі та ідеальним виходом, який необхідно отримати. Для успішного навчання мережі потрібно наблизити вихід мережі до бажаного виходу, т. Е. Послідовно зменшувати величину функції помилки. Це досягається налаштуванням між-нейронних зв'язків. Узагальнене дельта-правило навчає мережу шляхом обчислення функції помилки для заданого входу з подальшим її зворотним поширенням (ось звідки назва!) Від кожного шару до попереднього. Кожен нейрон в мережі має свої ваги, які налаштовуються, щоб зменшити величину функції помилки. Для нейронів вихідного шару відомі їхні фактичні та бажані значення виходів. Тому настройка ваг зв'язків для таких нейронів є відносно простою. Однак для

нейронів попередніх шарів настройка не настільки очевидна. Інтуїтивно ясно, що нейрони внутрішніх шарів, які пов'язані з виходами, що мають велику похибку, повинні змінювати свої ваги значно сильніше, ніж нейрони, з'єднані з майже коректними виходами. Іншими словами, ваги даного нейрона повинні змінюватися прямо пропорційно помилку тих нейронів, з якими даний нейрон пов'язаний. Ось чому зворотне поширення цих помилок через мережу дозволяє коректно налаштувати ваги зв'язків між усіма верствами. У цьому випадку величина функції помилки зменшується і мережа навчається.

Основні співвідношення методу зворотного поширення помилки отримані в [76] при наступних позначеннях:

E_p - величина функції помилки для образу p ;

t_{pj} - бажаний вихід нейрона j для образу p ;

o_{pj} - дійсний вихід нейрона j для образу p ;

w_{ij} - вага зв'язку між i -м та j -м нейронами.

Нехай функція помилки прямо пропорційна квадрату різниці між дійсним і бажаним виходами для всієї навчальної вибірки:

$$E_p = \frac{1}{2} \sum_j (t_{pj} - o_{pj})^2 \quad (2.14)$$

Множник $\frac{1}{2}$ вводиться для спрощення операції диференціювання.

Активація кожного нейрона j для образу p записується у вигляді зваженої суми:

$$net_{pj} = \sum_i w_{ij} o_{pi} \quad (2.15)$$

Вихід кожного нейрона j є значенням порогової функції f_j , яка активізується зваженою сумою. У багатошаровій мережі це зазвичай сігмоїдна функція, хоча може використовуватися будь-яка безперервно диференційована монотонна функція:

$$o_{pj} = f_j(\text{net}_{pj}) \quad (2.16)$$

Можна записати за правилом ланцюжка:

$$\frac{\partial E_p}{\partial w_{ij}} = \frac{\partial E_p}{\partial \text{net}_{pj}} \frac{\partial \text{net}_{pj}}{\partial w_{ij}} \quad (2.17)$$

Для другого співмножника в (2.18), використовуючи (2.16), отримуємо:

$$\frac{\partial \text{net}_{pj}}{\partial w_{ij}} = \frac{\partial}{\partial w_{ij}} \sum_k w_{kj} o_{pk} = \sum_k \frac{\partial w_{kj}}{\partial w_{ij}} o_{pk} = o_{pi} \quad (2.18)$$

оскільки, $\frac{\partial w_{kj}}{\partial w_{ij}} = 0$ за винятком випадку $k=i$, коли ця похідна дорівнює одиниці.

Зміна помилки як функція зміни входів нейрона визначається так:

$$-\frac{\partial E_p}{\partial \text{net}_{pj}} = \delta_{pj} \quad (2.19)$$

Тому (2.18) перетвориться до виду:

$$-\frac{\partial E_p}{\partial w_{ij}} = \delta_{pj} o_{pi} \quad (2.20)$$

Отже, зменшення величини E_p означає зміну ваги пропорційно $\delta_{pj} o_{pi}$:

$$\Delta_p w_{ij} = h \delta_{pj} o_{pi} \quad (2.21)$$

де h - коефіцієнт пропорційності, що впливає на швидкість навчання.

Тепер нам необхідно знати значення δ_{pj} для кожного нейрона. Використовуючи (2.20) і правило ланцюжка, можна записати:

$$\delta_{pj} = -\frac{\partial E_p}{\partial net_{pj}} = -\frac{\partial E_p}{\partial o_{pj}} \frac{\partial o_{pj}}{\partial net_{pj}} \quad (2.22)$$

Виходячи з (2.17), запишемо другий співмножник в (2.19):

$$\frac{\partial E_{pj}}{\partial net_{pj}} = f'_j(net_{pj}) \quad (2.23)$$

Тепер розглянемо перший співмножник в (2.22). Згідно (2.15), неважко отримати:

$$\frac{\partial E_p}{\partial o_{pj}} = -(t_{pj} - o_{pj}) \quad (2.24)$$

Тому

$$\delta_{pj} = f'_j(net_{pj})(t_{pj} - o_{pj}) \quad (2.25)$$

Останнє співвідношення є корисним для вихідних нейронів, оскільки для них відомі цільові та дійсні значення виходів. Однак для нейронів внутрішніх шарів цільові значення виходів не відомі.

Таким чином, якщо нейрон j - не вихідний нейрон, то знову використовуючи правило ланцюжка, а також співвідношення (2.16) і (2.20), можна записати:

$$\frac{\partial E_p}{\partial o_{pj}} = \sum_k \frac{\partial E_p}{\partial net_{pk}} \frac{\partial net_{pk}}{\partial o_{pj}} = \sum_k \frac{\partial E_p}{\partial net_{pk}} \frac{\partial}{\partial o_{pj}} \sum_i w_{ik} o_{pi} \quad (2.26)$$

$$\sum_k \frac{\partial E_p}{\partial net_{pk}} \frac{\partial}{\partial o_{pj}} \sum_i w_{ik} o_{pi} = - \sum_k \delta_{pk} w_{jk} \quad (2.27)$$

Тут сума по i зникає, оскільки приватна похідна не дорівнює нулю лише в одному випадку, так само як і в (2.19). Підставивши (2.27) в (2.23), отримаємо остаточний вираз:

$$\delta_{pj} = f_j'(net_{pj}) \sum_k \delta_{pk} w_{jk} \quad (2.28)$$

Рівняння (2.25) і (2.28) складають основу методу навчання багатошарової мережі.

Перевага використання сігмоїдної функції в якості нелінійного елемента полягає в тому, що вона дуже нагадує крокову функцію і, таким чином, може демонструвати поведінку, подібне природному нейрону. Сігмоїдна функція визначається як

$$f(net) = \frac{1}{1 + e^{-k \cdot net}}, \quad (2.29)$$

і має діапазон $0 < f < 1$. k - позитивна константа, що впливає на розтяг функції: збільшення k стискає функцію, а при $k \rightarrow \infty$ функція $f(net)$ наближається до функції Хевісайда. Цей коефіцієнт може використовуватися як параметр посилення, оскільки для слабких вхідних сигналів кут нахилу буде досить крутим і функція буде змінюватися досить швидко, виробляючи значне посилення сигналу. Для великих вхідних сигналів кут нахилу і, відповідно, посилення будуть набагато меншими. Це означає, що мережа може приймати великі сигнали і при цьому залишатися чутливою до слабких змін сигналу.

Проте головний сенс у використанні даної функції полягає в тому, що вона має просту похідну, і це значно полегшує застосування backpropagation-методу. Якщо вихід нейрона o_{pj} задається як

$$o_{pj} = f(net) = \frac{1}{1+e^{-k \cdot net}}, \quad (2.30)$$

то похідна по відношенню до даного нейрона $f'(net)$ обчислюється так:

$$f'(net) = \frac{ke^{-k \cdot net}}{(1+e^{-k \cdot net})^2} = \frac{kf(net)}{1-f(net)} = ko_{pj}(1 - o_{pj}), \quad (2.31)$$

т. е. є простою функцією від виходів нейронів.

Алгоритм налаштування нейронної мережі

Нижче наведено алгоритм настройки багатошарової нейронної мережі з використанням backpropagation-правила навчання. Для його застосування необхідно, щоб нейрони мали безперервно диференційовану нелінійну порогову функцію активації. Нехай це буде сігмоїдна функція

$$f(net) = \frac{1}{1+e^{-k \cdot net}}, \quad (2.32)$$

оскільки вона має просту похідну.

Алгоритм навчання полягає в наступному.

1. Задати початкові значення ваг і порогів кожного нейрона. Усім вагам і порогам присвоюються малі випадкові значення.
2. Уявити вхідний і вихідний образи з навчальної вибірки.

Нехай

$X_p = \{x_0, x_1, \dots, x_{n-1}\}$ – поточний вхідний образ,

$T_p = \{t_0, t_1, \dots, t_{m-1}\}$ – поточний вихідний образ з навчальної вибірки,

де n – число нейронів вхідного шару, m – число нейронів вихідного шару. При цьому $w_0 = -\Theta$ (зміщення) і $x_0 = b$.

При вирішенні задач класифікації образ може складатися з нулів, крім одного елемента, рівного 1, який і визначатиме клас поточного вхідного образу.

3. Розрахувати дійсні значення виходів.

Значення виходів нейронів кожного шару розраховуються як

$$y_{pj} = f\left[\sum_{i=0}^{n-1} w_i x_i\right] \quad (2.33)$$

і передаються на входи нейронів наступного шару. Вихідні значення нейронів вихідного шару рівні o_{pj} ,

4. Провести модифікацію ваг зв'язків.

Починаючи від вихідного шару i , рухаючись у зворотному напрямку, необхідно змінювати ваги зв'язків таким чином:

$$w_{ij}(t + 1) = w_{ij}(t) + h\delta_{pj}o_{pj}, \quad (2.34)$$

де w_{ip} - вага зв'язку між i -м та j -м нейронами на t -му кроці;

h - швидкість навчання;

δ_{pj} - швидкість зміни помилки для нейрона j при пред'явленні образу p .

Для нейронів вихідного шару

$$\delta_{pj} = k o_{pj} (1 - o_{pj}) (t_{pj} - o_{pj}), \quad (2.35)$$

для нейронів внутрішніх шарів

$$\delta_{pj} = k o_{pj} (1 - o_{pj}) \sum_k \delta_{pk} w_{jk}, \quad (2.36)$$

де під знаком суми стоять величини, що відносяться до нейронів наступного шару.

2.1.3 Генетичний алгоритм

Генетичний алгоритм (англ. genetic algorithm) — це еволюційний алгоритм пошуку, що використовується для вирішення задач оптимізації і моделювання шляхом послідовного підбору, комбінування і варіації шуканих параметрів з використанням механізмів, що нагадують біологічну еволюцію.

Особливістю генетичного алгоритму є акцент на використання оператора «схрещення», який виконує операцію рекомбінації рішень-кандидатів, роль якої аналогічна ролі схрещення в живій природі. «Батьком-засновником» генетичних алгоритмів вважається Джон Голланд (англ. John Holland), книга якого «Адаптація в природних і штучних системах» (англ. *Adaptation in Natural and Artificial Systems*) є фундаментальною в цій сфері досліджень.

Задача кодується таким чином, щоб її вирішення могло бути представлено в вигляді масиву подібного до інформації складу хромосоми. Цей масив часто називають саме так «хромосома». Випадковим чином в масиві створюється деяка кількість початкових елементів «осіб», або початкова популяція. Особи оцінюються з використанням функції пристосування, в результаті якої кожній особі присвоюється певне значення пристосованості, яке визначає можливість виживання особи. Після цього з використанням отриманих значень пристосованості вибираються особи, допущені до схрещення (селекція). До осіб застосовується «генетичні оператори» (в більшості випадків це оператор схрещення (crossover) і оператор мутації (mutation)), створюючи таким чином наступне покоління осіб. Особи

наступного покоління також оцінюються застосуванням генетичних операторів і виконується селекція і мутація. Так моделюється еволюційний процес, що продовжується декілька життєвих циклів (поколінь), поки не буде виконано критерій зупинки алгоритму. Таким критерієм може бути:

1. знаходження глобального, або надоптимального вирішення;
2. вичерпання числа поколінь, що відпущені на еволюцію;
3. вичерпання часу, відпущеного на еволюцію.

Генетичні алгоритми можуть використати для пошуку рішень в дуже великих і важких просторах пошуку.

Зразу ж відзначимо, що застосування генетичного алгоритму в даній задачі є недоцільним, тому що критерій добору хромосом і використання процедур є евристичним, що зовсім не гарантує відшукання найкращого рішення. Іншим недоліком є велика обчислювана складність.

2.2 Етапи інформаційної технології, які використовуються при побудові моделі ТПБК.

Етап 1. Побудова дерева висновків і фазифікації впливаючих факторів

Дерево виведення - це граф, структура якого відображає класифікацію факторів, що впливають на прогнозований показник, - вихід біогазу. Вершини дерева інтерпретуються таким чином: корінь дерева - вихідний показник, термінальні вершини - впливаючі фактори, нетермінальні вершини - класифікаторів, що впливають.

Етап фазифікації передбачає вибір нечітких термів для лінгвістичної оцінки факторів і формалізацію цих термів за допомогою функцій належності.

Етап 2. Побудова нечітких матриць знань.

Нечітка матриця знань є аналогом матриці планування експерименту [10]. Але на відміну від реальних експериментальних даних, в нечітку матрицю знань заносяться результати так званого віртуального а точніше уявного експерименту. У процесі такого експерименту експерт відповідає на питання: якою буде лінгвістична оцінка вихідного показника при заданому поєднанні лінгвістичних оцінок факторів. Ієрархічний характер дерева виведення передбачає необхідність побудови нечітких матриць знань для всіх нетермінальних вершин і кореня цього дерева.

Етап 3. Нечіткий логічний висновок і дефазифікація вихідного показника.

Техніка нечіткого логічного висновку, застосована до інформації, зібраної на попередніх етапах, дозволяє обчислити показник, який прогнозується, у вигляді нечіткої множини. Це нечітка множина визначає рівень виходу біогазу для фіксованого вектора факторів, що впливають. Щоб перейти від отриманого нечіткої множини до кількісної оцінки потрібно виконати процедуру дефазифікації, (перетворення нечіткої інформації в чітку форму). Серед різних методів дефазифікації найбільш поширеним є знаходження "центру мас" плоскої фігури, обмеженої функцією приналежності нечіткої множини і горизонтальною координатою. Модель нечіткого логічного висновку разом з процедурою дефазифікації забезпечує можливість спостереження за зміною вихідного показника при варіації факторів, що впливають на кінцевий результат.

2.3 Дерево виведення рішень

Дерево виведення, що визначає послідовності розрахунків при прогнозуванні кількості біогазу, показано на рис. 2.6.

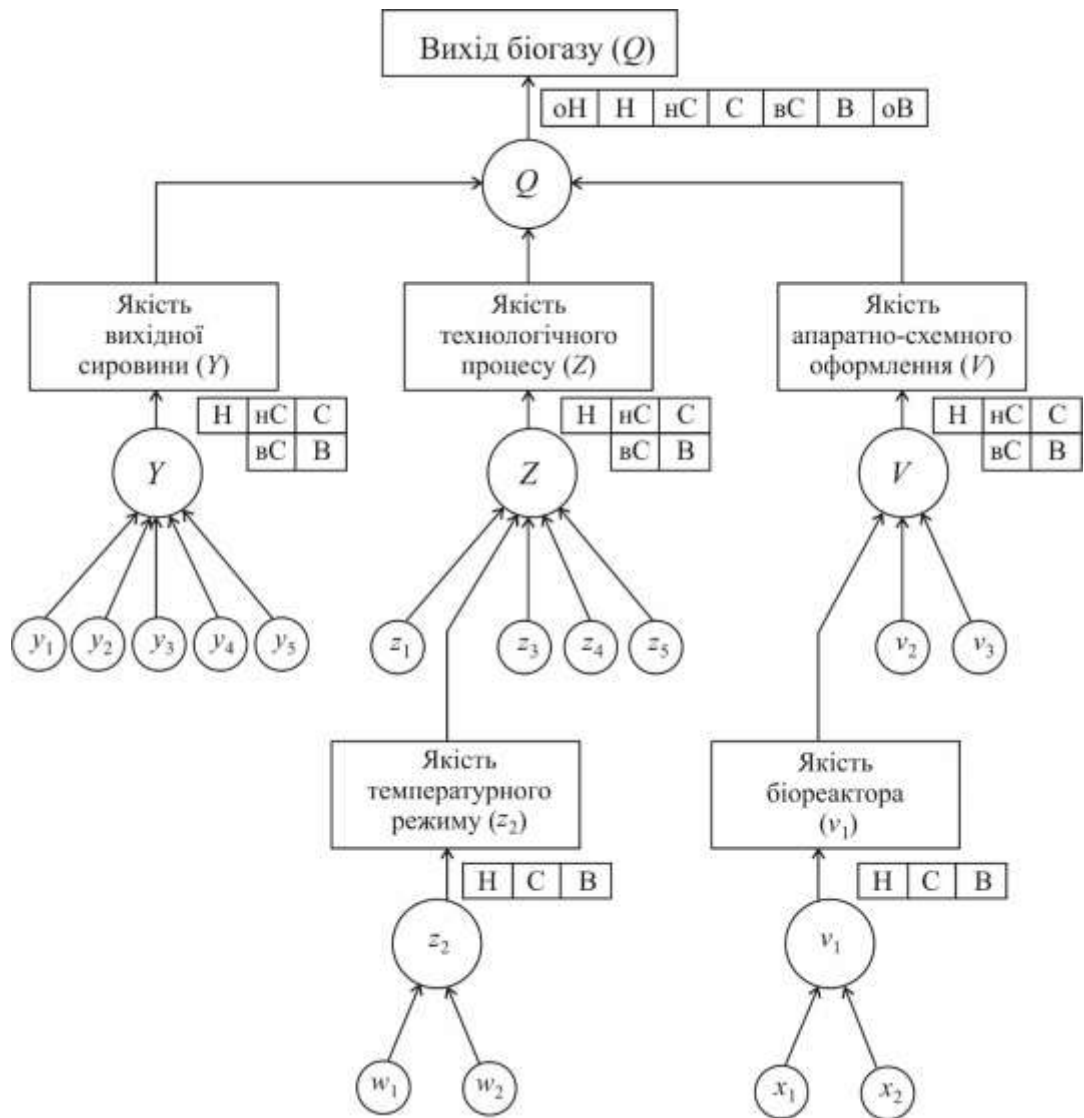


Рисунок 2.6 – Дерево виведення показника кількості біогазу

Одинарними кружками позначені впливають фактори, які наведені в табл. 2.2. Вони розглядаються як лінгвістичні змінні, задані на відповідних універсальних множинах за допомогою нечітких термів. Функції належності нечітких термів, наведених в табл. 2.2, отримані методом парних порівнянь.

Таблиця 2.2 - Фактори як лінгвістичні змінні

Позначення і назва змінної	універсальне множина	Терми для оцінки
Y_1 - якість сировини	(5,13) у.о.	корови (1),

		бички (2), свині (3), кури (4), індики (5)
У ₂ - вологість сировини	(80,98)%	низька (1), нормальна (2), висока (3)
У ₃ - гранулометричний склад	(1,2) у.о.	подрібнена (1), неподрібнена (2)
У ₄ - кількість жирних кислот	(600,1500) мг / л	норма (1), багато (2)
У ₅ - водневий показник	(6.5,8.5) PH	кислий (1), нейтральний (2), лужної (3)
z ₁ - час зброджування	(5,15) діб	мале (1), нормальне (2), підвищений (3)
w ₁ - температура бродіння	(15,55) ° C	психрофільні (1), мезофільні (2), термофільний (3)
w ₂ - стабільність температурного режиму	(0,0.3) Dt / t	низька (1), задовільна (2), висока (3)
z ₃ - якість перемішування	(0,4) у.о.	відсутнє (1), мінімальне (2), звичайне (3)
z ₄ - вміст метаногенної мікрофлори	(0.1,41010) кліть / Г (СВ)	низьке (1), середнє (2), висока (3)

z_5 - доза завантаження реактора	(6,20)%	низька (1), середня (2), висока (3)
x_1 - тип біореактора	(1,1.5) у.о.	примітивна (1), традиційна (2), модернізована (3)
x_2 - ступінь іммобілізації	(1,3) у.о.	незначна (1), достатня (2), висока (3)
v_2 - якість додаткового обладнання	(0,100) балів	низьке (1), середнє (2), висока (3)
v_3 - якість апаратно-схемного оформлення	(0,100) балів	низьке (1), середнє (2), висока (3)

У табл. 2.2 використані наступні позначення: у.о. - умовні одиниці, Dt/t - відношення найбільшого відхилення температури до заданого рівня, РН - водневий показник, клітин / Г (СР) - клітин на грам сухої речовини. Для факторів v_2 і v_3 , що характеризують якість біореактора, користуємось стобальною шкалою інтегральної оцінки. Але кожен з цих факторів може розглядатися на більш глибоких рівнях деталізації.

2.4 Функції належності вхідних лінгвістичних термів

За визначенням [5], функція приналежності характеризує суб'єктивну міру (в діапазоні $[0,1]$) впевненості експерта в тому, що чітке значення відповідає нечіткому терму. Найбільшого поширення в практичних застосуваннях [6] отримали трикутні, трапецієподібні і дзвоновидні (гаусові) функції приналежності, параметри яких дозволяють змінювати форму функцій.

Ми будемо використовувати просту і зручну для налаштування аналітичну модель функцій приналежності змінної нечіткому терму у вигляді:

$$\mu^T = \frac{1}{1 + \left(\frac{x-b}{c}\right)^2}, \quad (2.37)$$

Де b і c - параметри налаштування: b - координата точки максимуму функції,

$\mu^T(b) = 1$; c - коефіцієнт концентрації - розтягування функції (рис. 2.2).

Для нечіткого терма T число b представляє найбільш можливе значення змінної x .

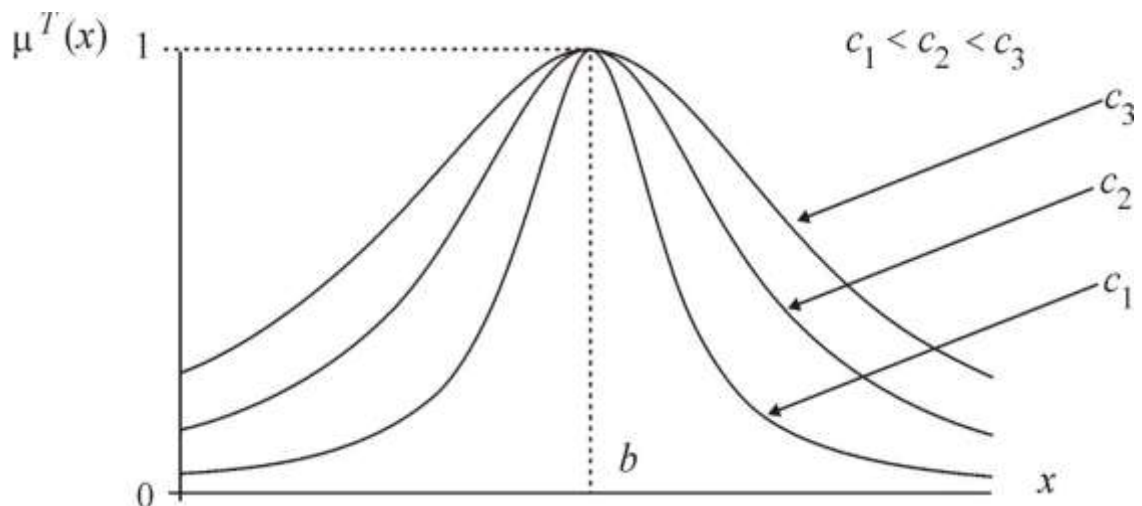


Рисунок 2.7 – Модель функції належності

Для моделювання використовувалися дзвоноподібні функції приналежності, задані на єдиній універсальній множині $U = [0,100]$, з параметрами центру (b) і стиснення-розтягування (c), представленими в табл. 2.2.

Таблиця 2.3 - Параметри функцій належності

Терм	ДН	Н	НС	С	ВС	В	ДВ
b	0	16,7	33,3	50	66,7	83,3	100
c	15	15	15	15	15	15	15

Нижче наведені функції приналежності лінгвістичних термів всіх вхідних параметрів Рис.2.8 - 2.22

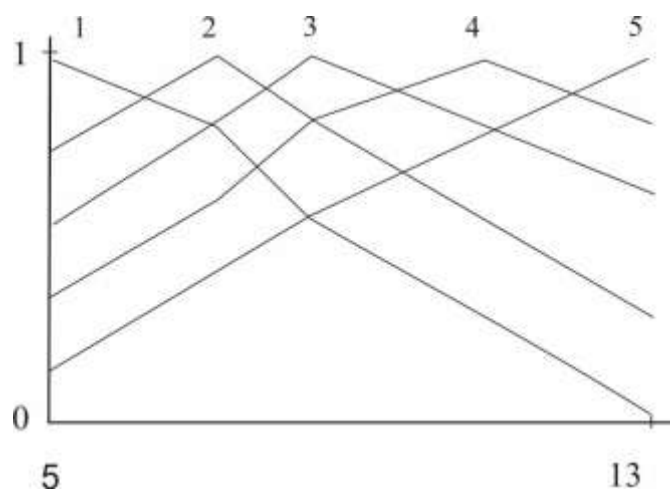


Рисунок 2.8 – Функції належності лінгвістичних термів, що характеризують y_1 - якість сировини

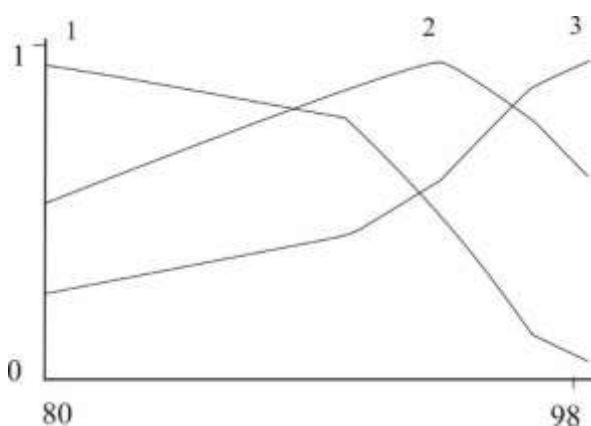


Рисунок 2.9 – Функції належності лінгвістичних термів, що характеризують y_2 - вологість сировини

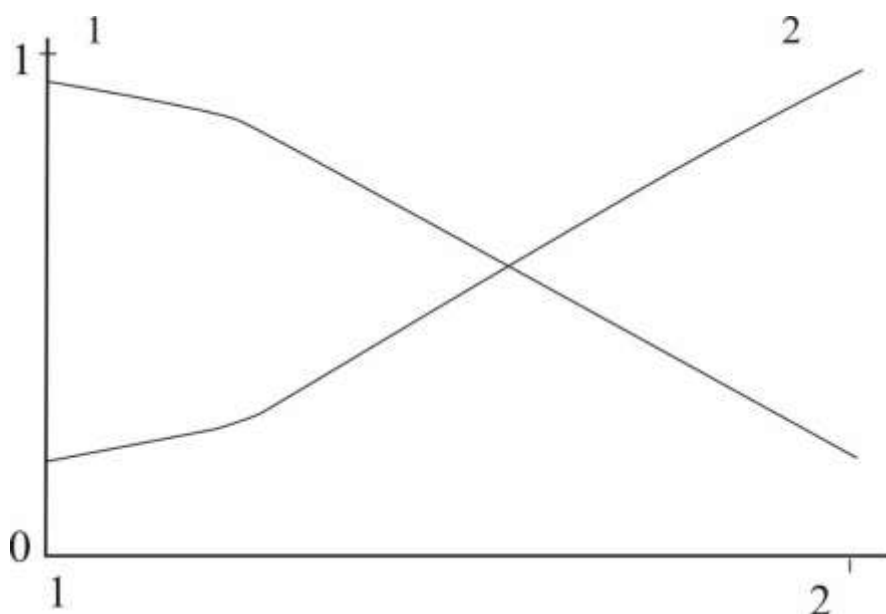


Рисунок 2.10 – Функції належності лінгвістичних термів, що характеризують u_3 - гранулометричний склад сировини

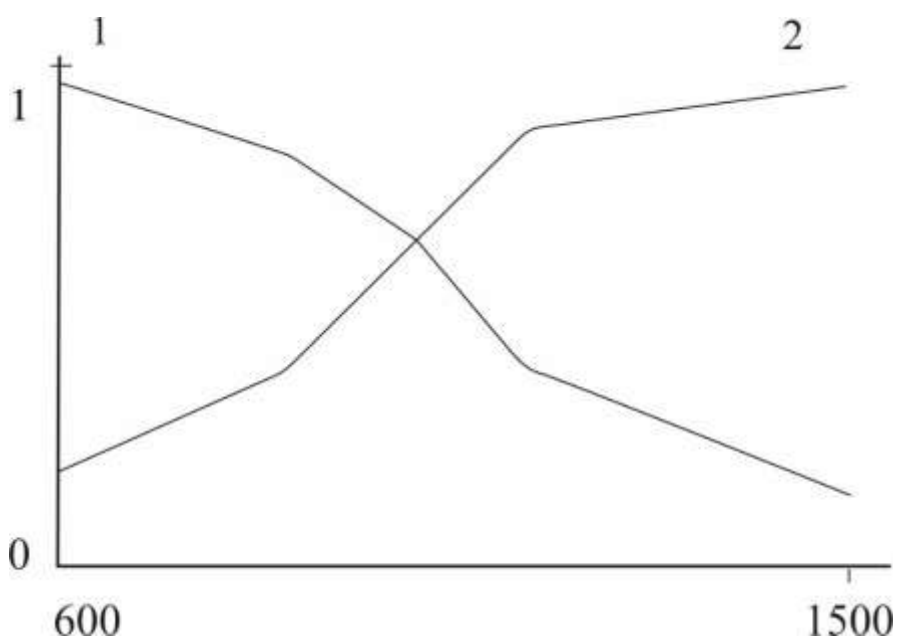


Рисунок 2.11 – Функції належності лінгвістичних термів, що характеризують u_4 – кількість жирних кислот

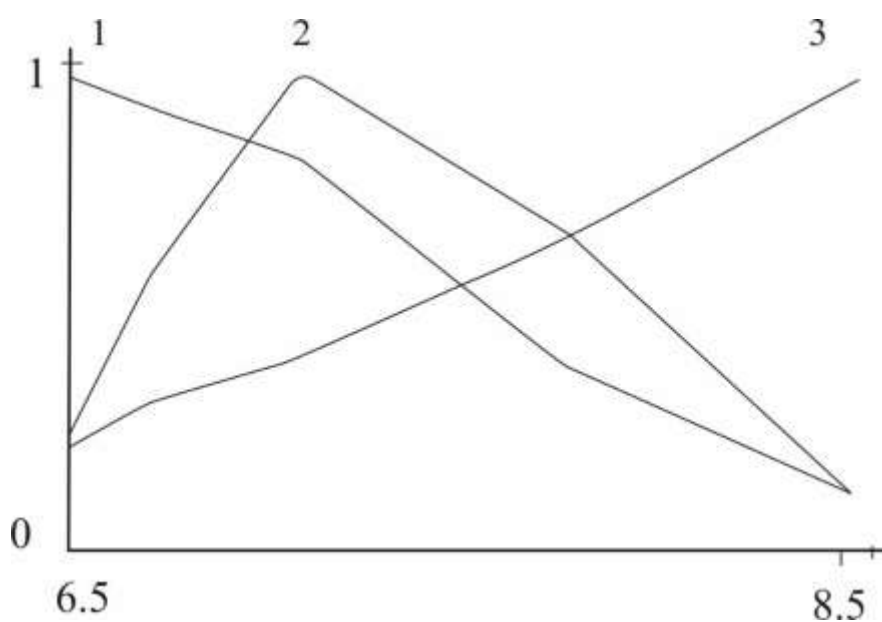


Рисунок 2.12 – Функції належності лінгвістичних термів, що характеризують y_5 – водневий показник рН

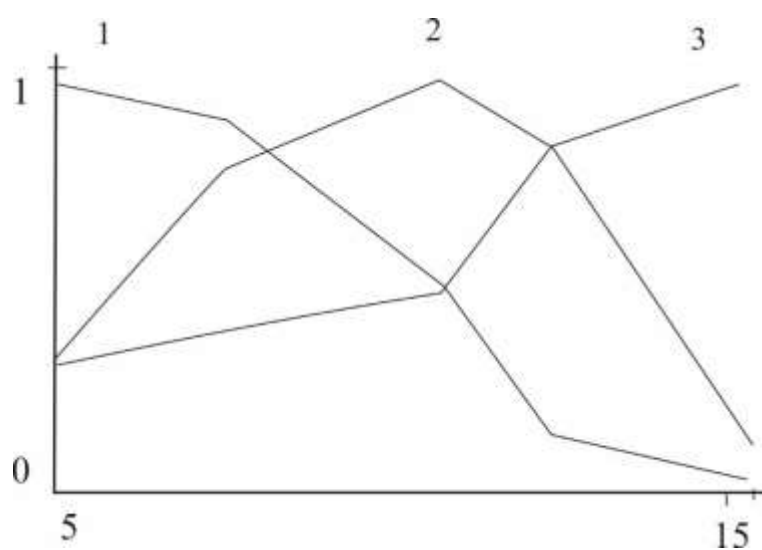


Рисунок 2.13 – Функції належності лінгвістичних термів, що характеризують z_1 – час зброджування

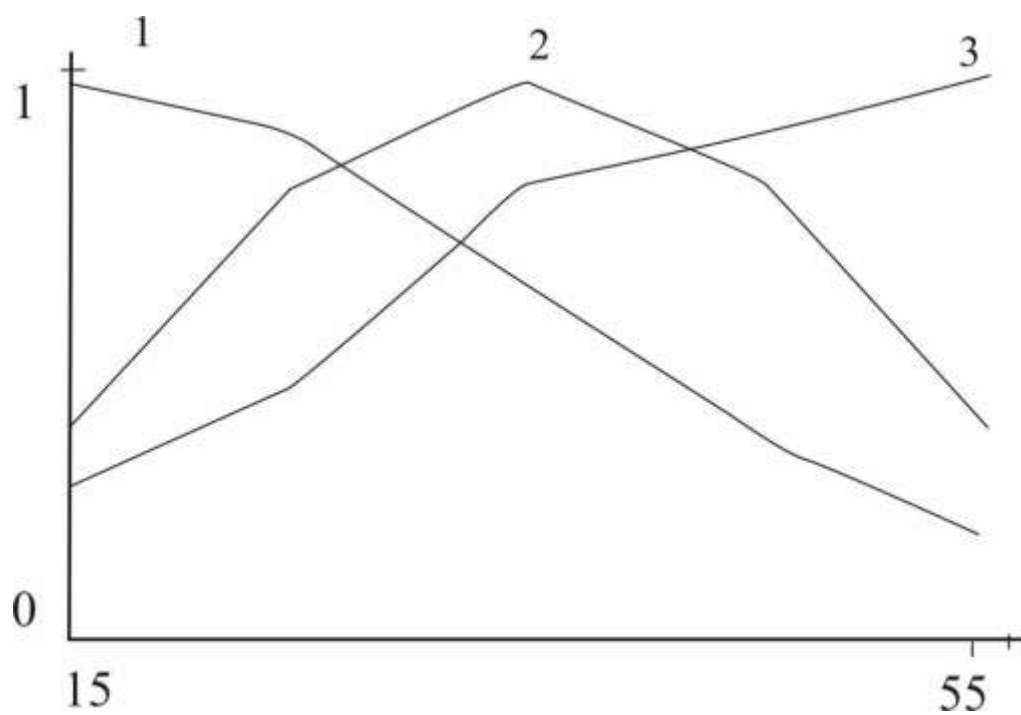


Рисунок 2.14 – Функції належності лінгвістичних термів, що характеризують w_1 – температуру зброджування

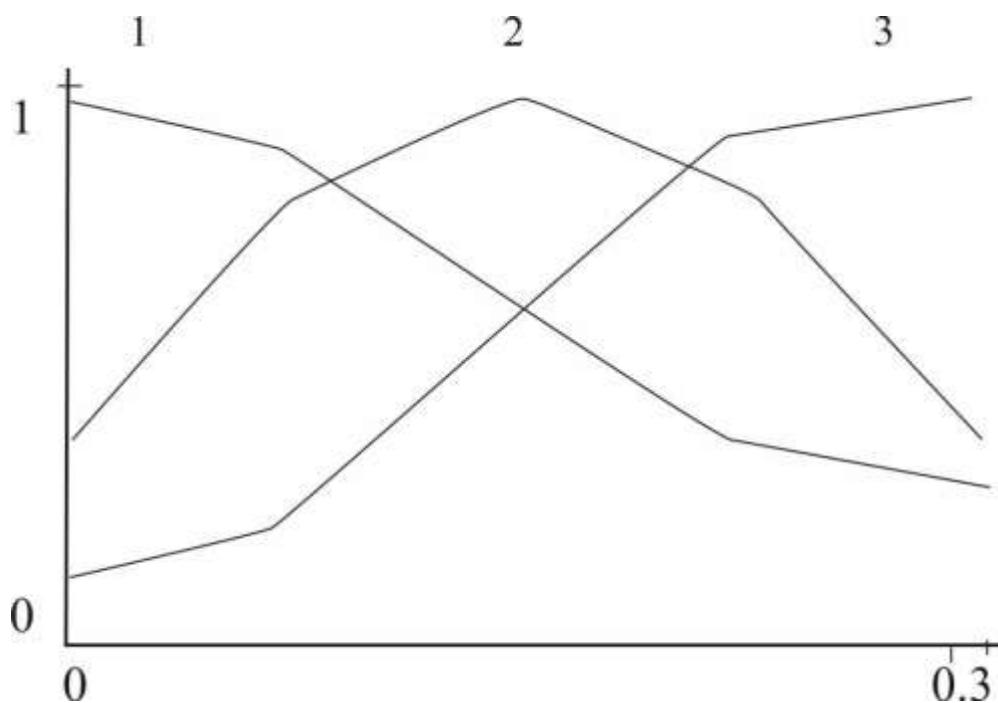


Рисунок 2.15 – Функції належності лінгвістичних термів, що характеризують w_2 – стабільність температурного режиму

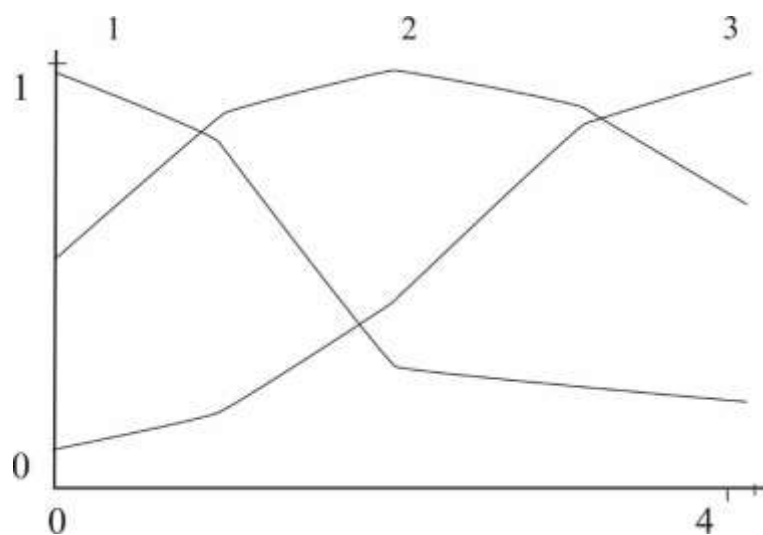


Рисунок 2.16 – Функції належності лінгвістичних термів, що характеризують z_3 - якість перемішування

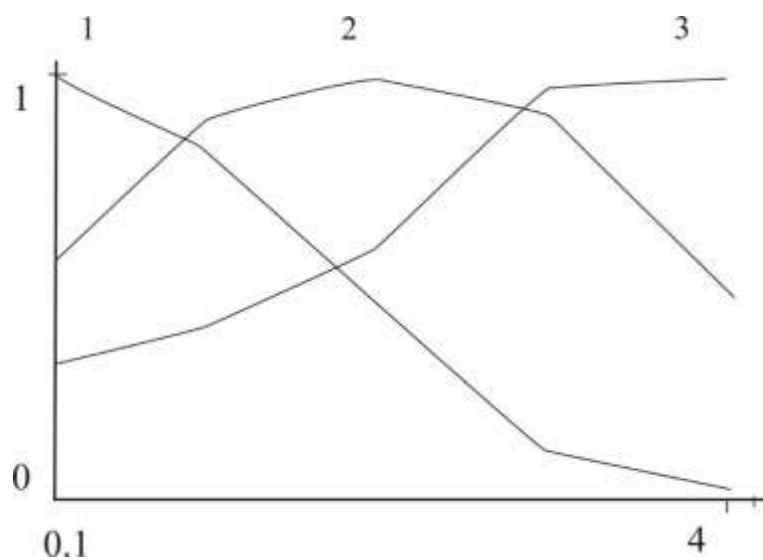


Рисунок 2.17 – Функції належності лінгвістичних термів, що характеризують z_4 – склад метаногенної мікрофлори

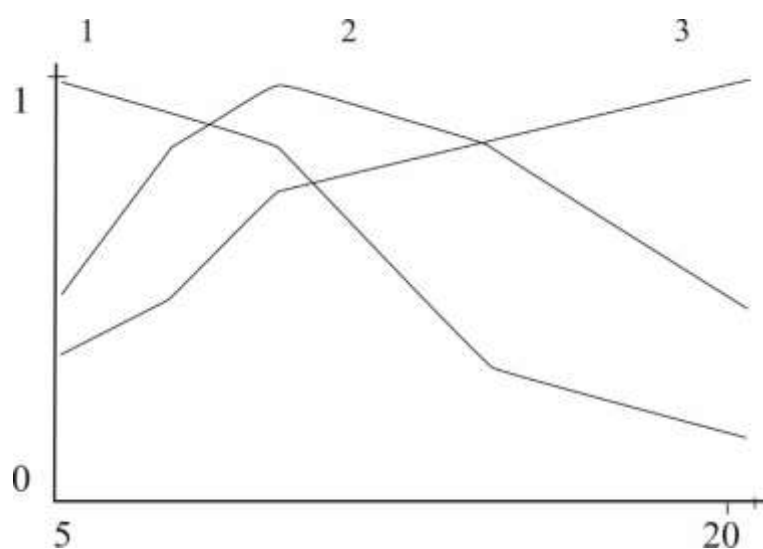


Рисунок 2.18 – Функції належності лінгвістичних термів, що характеризують z_5 – доза завантаження реактора

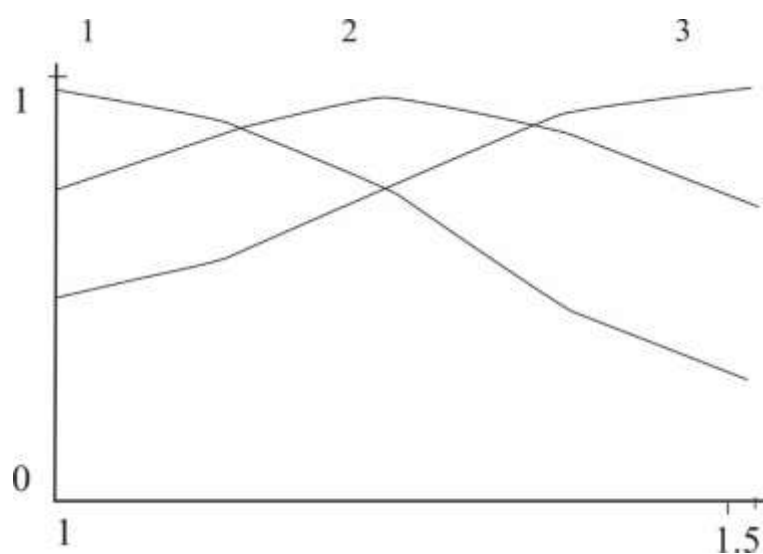


Рисунок 2.19 – Функції належності лінгвістичних термів, що характеризують x_1 – тип біореактора

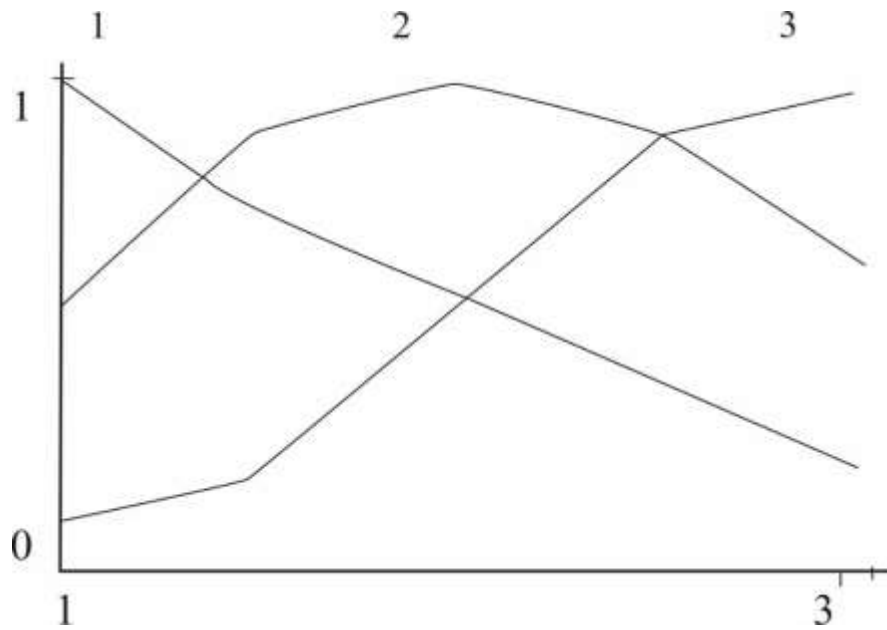


Рисунок 2.20 – Функції належності лінгвістичних термів, що характеризують x_2 – ступінь іммобілізації

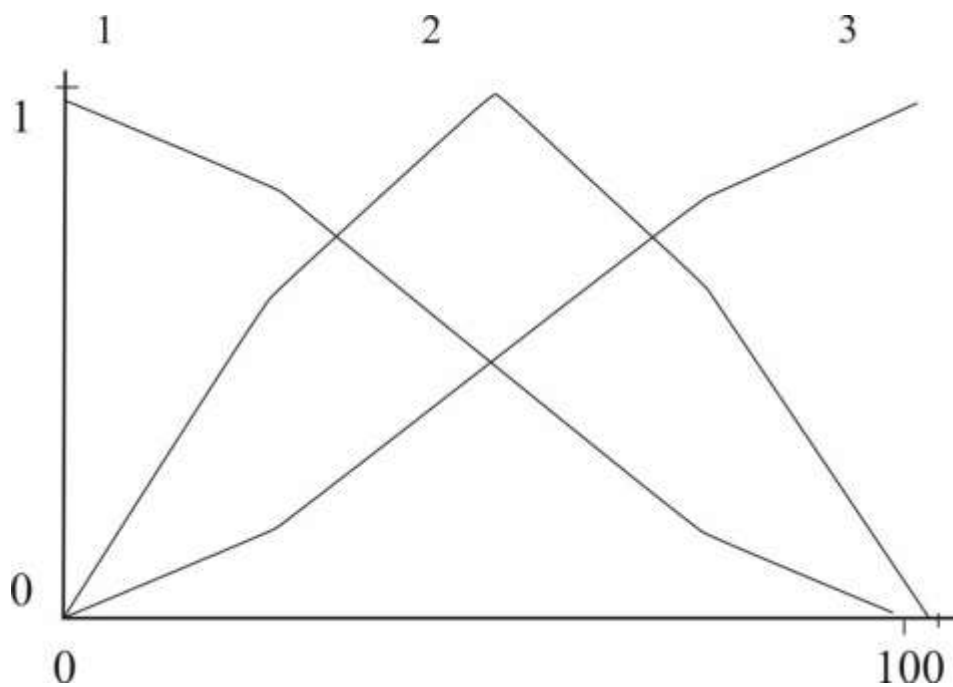


Рисунок 2.21 – Функції належності лінгвістичних термів, що характеризують v_2 – якість додаткового обладнання

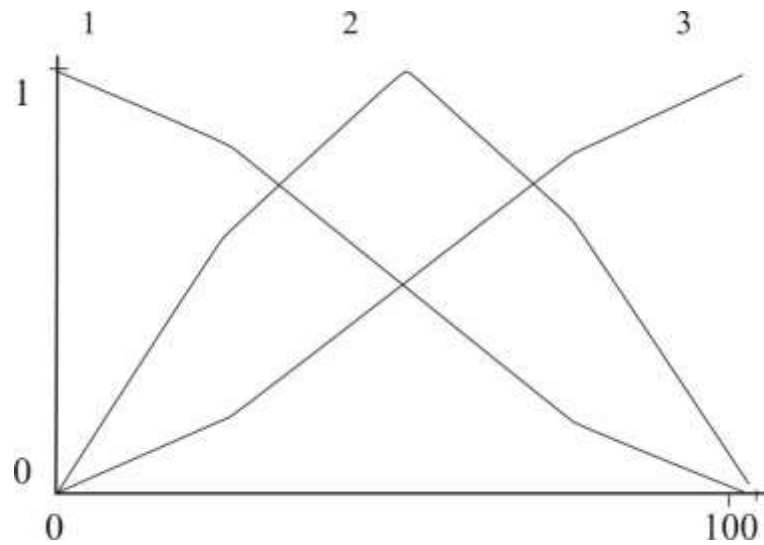


Рисунок 2.22 – Функції належності лінгвістичних термів, що характеризують v_3 – якість апаратно-схемного оформлення

2.5 Моделювання та розробка бази знань

Запишемо основні залежності для вузлів дерева рішень

$$Q = f_Q(Y, Z, V), \quad (2.38)$$

$$Y = f_Y(y_1, y_2, y_3, y_4, y_5), \quad (2.39)$$

$$Z = f_Z(z_1, z_2, z_3, z_4, z_5), \quad (2.40)$$

$$V = f_V(v_1, v_2, v_3), \quad (2.41)$$

$$Z_2 = (w_1, w_2), \quad (2.42)$$

$$V_1 = (x_1, x_2), \quad (2.43)$$

Для представлення цих залежностей лінгвістичними продукційними правилами «ЯКЩО – ТО», вводяться нечіткі терми-оцінки змінних, що стоять в лівих частинах залежностей (2.38) – (2.43). на дереві виводу (Рис.2.6) ці терми приведені поруч із стрілками, що виходять із відповідних вершин-вузлів.

Цей етап називається фазифікацією. Наприклад для вихідної змінної величини Q <вихід біогазу> маємо слідуючі значення лінгвістичних термів:

- ДН - <дуже низький>;
- Н - <низький>;
- НС - <нижче середнього>;
- С - <середній>;
- ВС - <вище середнього>;
- В - <високий>;
- ДВ - <дуже високий>.

Для змінної Y <якість сировини>, відповідно:

- Н - <низький>;
- НС - <нижче середнього>;
- С - <середній>;
- ВС - <вище середнього>;
- В - <високий>;

Для змінної Z <якість технології>, відповідно:

- Н - <низький>;
- НС - <нижче середнього>;
- С - <середній>;
- ВС - <вище середнього>;
- В - <високий>.

Для змінної V <якість апаратури>, присвоїмо значення:

- Н - <низький>;
- НС - <нижче середнього>;
- С - <середній>;
- ВС - <вище середнього>;
- В - <високий>.

Фрагмент нечіткої матриці знань, що буде використана для моделювання залежності (2.38), представлена в таблиці 2.4

Таблиця 2.4 – Фрагмент бази знань (мтричне представлення)

ЯКЩО			ТО
Якість сировини (Y)	Якість технологічного процесу (Z)	Якість апаратно-схемного оформлення (V)	Вихід біогазу (Q)
низький	низький	низький	дуже низький
низький	низький	нижче середнього	
низький	нижче середнього	нижче середнього	
низький	низький	високий	низький
нижче середнього	нижче середнього	нижче середнього	
низький	низький	середній	
нижче середнього	нижче середнього	низький	нижче середнього
нижче середнього	нижче середнього	нижче середнього	
нижче середнього	нижче середнього	середній	
середній	нижче середнього	низький	середній
середній	середній	середній	
середній	середній	вище середнього	
середній	середній	високий	вище середнього
вище середнього	вище середнього	вище середнього	
середній	вище середнього	вище середнього	
середній	вище середнього	середній	високий
середній	вище середнього	вище середнього	

вище середнього	вище середнього	вище середнього	
вище середнього	вище середнього	високий	дуже високий
високий	вище середнього	високий	
високий	високий	високий	

Приклади експертних висловлювань, їх ще називають продукційними правилами або логічними ланцюжками прийняття рішень, що стануть наповненням бази знань, будуть висловлювання типу:

ЯКЩО якість сировини $Y = N(\text{низька})$,

І якість технології $Z = N(\text{низька})$,

І якість апаратури $V = N(\text{низька})$,

АБО якість сировини $Y = N(\text{низька})$,

І якість технології $Z = N(\text{низька})$,

І якість апаратури $V = N(\text{вище середнього})$,

АБО якість сировини $Y = N(\text{низька})$,

І якість технології $Z = N(\text{нижче середнього})$,

І якість апаратури $V = N(\text{нижче середнього})$,

ТО вихід біогазу $Q = ДН(\text{дуже низький})$.

ЯКЩО якість сировини $Y = N(\text{нижче середнього})$,

І якість технології $Z = N(\text{нижче середнього})$,

І якість апаратури $V = N(\text{нижче середнього})$,

АБО якість сировини $Y = N(\text{нижче середнього})$,

І якість технології $Z = N(\text{нижче середнього})$,

І якість апаратури $V = N(\text{середній})$,

АБО якість сировини $Y = N(\text{нижче середнього})$,

І якість технології $Z = N(\text{середній})$,

І якість апаратури $V = N(\text{середній})$,

ТО вихід біогазу $Q = ДН(\text{низький})$

2.6 Висновки

В другому розділі магістерської кваліфікаційної роботи виконані розробка та моделювання етапів удосконаленої технології. При цьому проведені наступні дослідження:

- Проведено обґрунтування вибору методології моделювання роботи інформаційної технології, де проаналізована можливість застосування, в якості методології моделювання, нечіткої логіки, нейронних мереж та генетичних алгоритмів. У якості базової методології вибрана нечітка логіка.
- З використанням апарату нечіткої логіки розроблена модель роботи інформаційної технології управління біоконверсією;
- Виконана формалізація моделі (визначено входи і виходи об'єкта моделювання).
- Виконана фазифікація моделі (визначено лінгвістичні змінні та їх функції належності).
- Розроблена нечітка база знань.
- Виконано налаштування моделі з використанням експериментальної тестової вибірки.

Використовуючи удосконалену модель нової технології, ми можемо приступити до проектування експериментального модуля, що її реалізує.

3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ УПРАВЛІННЯ БІОКОНВЕРСІЄЮ.

3.1 Обґрунтування та вибір середовища та мови програмування

Оптимальним рішенням для реалізації програмної частини магістерської кваліфікаційної роботи є вибір середовища програмування Visual Studio 2016. Віддавши перевагу цьому середовищу ми можемо, по перше, скористатись безоплатною ліцензією, наданою студентам ВНТУ фірмою ISBN, по друге використати вбудовану CASE – технологію та універсальну мову моделювання UML, і по третє, скориставшись можливостями компонентної технології використати при реалізації шаблон проектування систем з нечіткою логікою FUZZY EXPERIENCE.

Microsoft Visual Studio - інтегроване середовище розробки програмного забезпечення і ряд інших інструментальних засобів. Дані продукти дозволяють розробляти як консольні додатки, так і додатки з графічним інтерфейсом, в тому числі з підтримкою технології Windows Forms, а також веб-сайти, веб-додатки, веб-служби як в рідному, так і в управляемому кодах для всіх платформ, підтримуваних Windows, Windows Mobile, Windows CE, .NET Framework, Xbox, Windows Phone .NET Compact Framework і Silverlight. Visual Studio включає в себе редактор вихідного коду з підтримкою технології IntelliSense і можливістю найпростішого рефакторінга коду. Вбудований відладчик може працювати як відладчик рівня вихідного коду, так і відладчик машинного рівня. Решта вбудованих інструментів включають в себе редактор форм для спрощення створення графічного інтерфейсу додатку, веб-редактор, дизайнер класів і дизайнерсхеми бази даних. Visual Studio дозволяє створювати і підключати сторонні додатки (плагіни) для розширення функціональності практично на кожному рівні, включаючи додавання підтримки системконтроля версій вихідного коду (як, наприклад, Subversion і Visual SourceSafe), додавання нових наборів інструментів (наприклад, для редагування і

візуального проектування коду на предметно-орієнтованих мовах програмування) або інструментів для інших аспектів процесу розробки програмного забезпечення (наприклад, клієнт Team Explorer для роботи з Team Foundation Server

Оскільки було прийнято рішення по використанні середовища Visual Studio 2016, то найбільш прийнятним рішенням щодо мови програмування є C#.

І нарешті останнє, оскільки ми використовуємо середовище програмування Visual Studio 2016 для реалізації середовища прийняття рішень засобами нечіткої логіки то було б доречно визначити Visual Studio 2016 як мета середовище.

3.2 Декомпозиція програмного забезпечення та розробка структури компонентів.

В результаті аналізу проведеного в розділах 1,2 представимо діаграму компонентів так, як це показано на рис.3.1



Рисунок 3.1 – Діаграма компонентів модуля

Для реалізації бази знань та її налаштування використаємо середовище автоматизованого проектування експертних систем на основі нечіткої логіки

«FUZZY EXPERIENS». Структура середовища зображена на рис. 3.2. Вона містить 9 програмних блоків, 2 з яких складають власне експертну систему, а решта, - середовище розробки нечітких експертних систем. Пояснимо їх призначення.

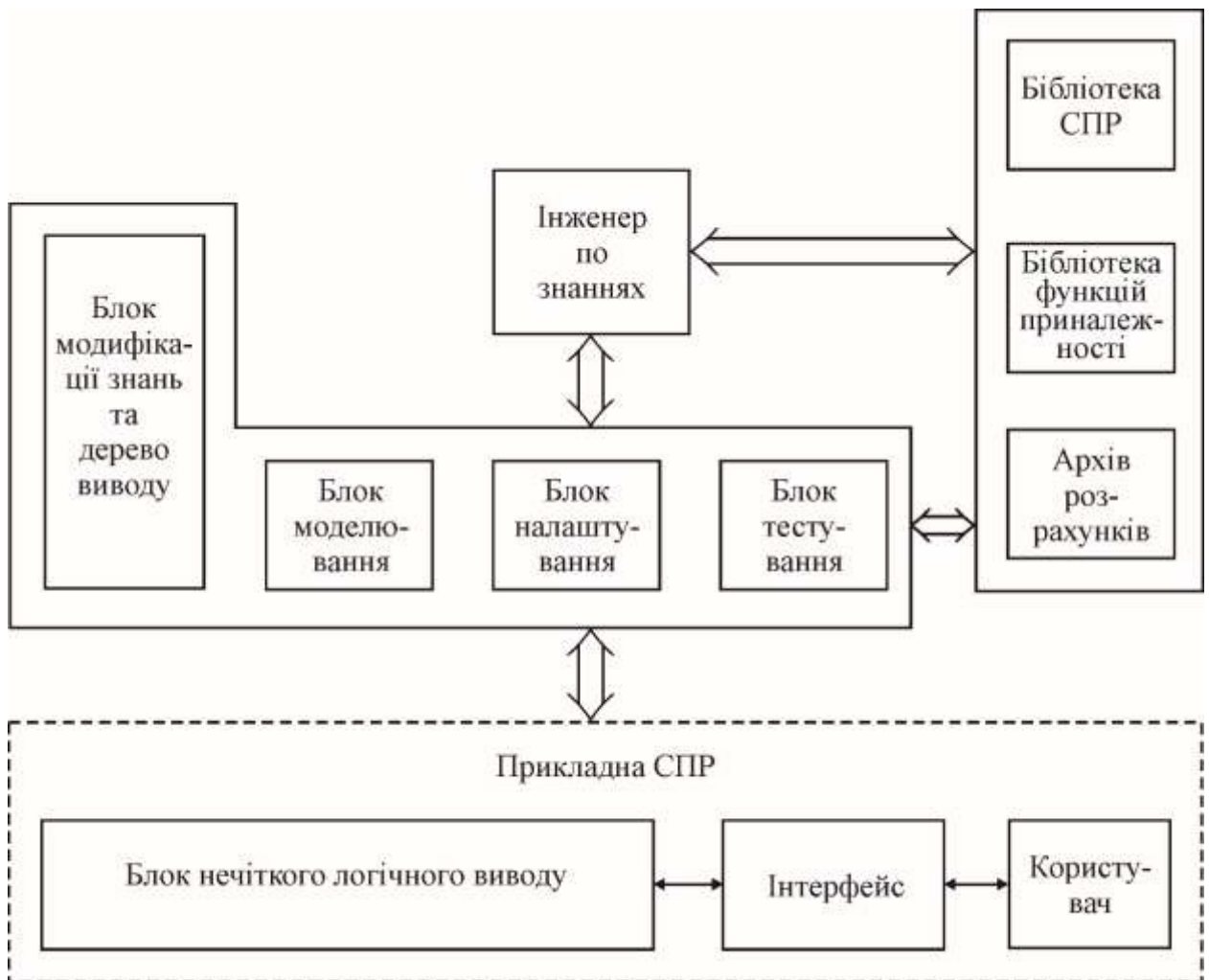


Рисунок 3.2 – Структурна схема середовища прийняття рішень

Блок модифікації нечіткої бази знань і дерева виводу містить всі екранні форми, необхідні для внесення змін в блоки експертної системи: дерево логічного висновку, базу знань і функції належності. Це основний блок середовища розробки нечітких баз знань. Він виконує наступні функції:

- а) формування дерева логічного виводу;
- б) задання функцій належності лінгвістичних термів:
 - Модифікованим методом Сааті,
 - Методом статистичної обробки експертної інформації,
- в) заповнення нечітких баз знань;
- г) збереження (витяг) варіантів експертних систем з бібліотеки.

Блок моделювання використовується для отримання графіків і поверхонь, що відбивають залежність вихідної змінної від однієї або двох вхідних змінних при фіксованих значеннях інших змінних. Мета подібного моделювання полягає в дослідженні поведінки об'єкта в різних областях факторного простору.

Блок налаштування призначений для вирішення завдань оптимізації нечіткої бази знань з метою підвищення якості ідентифікації нелінійних об'єктів. Цей блок виконує наступні функції:

- а) запит навчальної вибірки;
- б) вирішення задач оптимізації нечітких баз знань градієнтним методом;
- в) вирішення завдань оптимізації нечітких баз знань з параметричними функціями належності за допомогою генетичного алгоритму;
- г) рішення задач оптимізації нечітких баз знань з α -рівневими функціями належності за допомогою генетичного алгоритму.

Блок тестування експертної системи призначений для виконання наступних дій:

- а) запит тестуючої вибірки;
- б) оцінка якості ідентифікації в точках тестируючої вибірки.

Блок документування здійснює видачу інформації про побудовану експертну систему у вигляді, зручному для інженера по знаннях.

Бібліотека експертних систем використовується для зберігання створених експертних систем на різних етапах їх розробки (до налаштування і після налаштування).

Бібліотека функцій належності містить набір стандартних моделей функцій належності:

- а) трапецієподібні;
- б) трикутні;
- в) дзвоноподібні (з параметрами b і c);
- г) експоненціальні.

Архів розрахунків дозволяє зберігати результати виконуваних розрахунків на різних етапах створення і функціонування системи.

3.3 Розробка діаграми класів

Діаграма класів представлена на рис. 3.3 Вона складається із 13 слідуючих класів: **FUNCT**, **TERM**, **HIGH**, **HIGHM**, **MIDDLE**, **LOWM**, **LOW**, **CYCLE**, **CYCLE1**, **CYCLE2**, **CYCLE3**, **CYCLE4**, **FCYCLE**.

Клас **FUNCT**

Призначення.

Клас **FUNCT** зберігає бібліотеку функцій належності логічних термів. Призначений для генерації через наслідування класів **HIGH**, **HIGHM**, **MIDDLE**, **LOWM**, **LOW**.

Відповідальність:

Навчання (тонке налаштування функцій належності логічних термів)

Формування бази знань.

Атрибути.

Зберігає параметри функцій належності різних форм для кожного із дінгвістичних термів.

Методи.

Приведення функцій належності до єдиного універсального інтервалу. Поповнення та редагування бібліотеки функцій належності. Вибірки та корекції функцій.

Клас **TERM** – абстрактний клас для створення класів **HIGH, HIGHM, MIDDLE, LOWM, LOW**.

Класи **HIGH, HIGHM, MIDDLE, LOWM, LOW**

Призначення.

Класи **HIGH, HIGHM, MIDDLE, LOWM, LOW** призначені для зберігання параметрів функцій належності, відповідно до кожного із логічних термів. Успадковує параметри та методи класу **FUNCT**.

Клас **CYCLE**

Призначення.

Абстрактний клас для створення класів **CYCLE1, CYCLE2, CYCLE3, CYCLE4**.

Атрибути.

Кількість циклів та кількіість років у циклі.

Класи **CYCLE1, CYCLE2, CYCLE3, CYCLE4**

Призначення.

Зберігають мережу залежностей в межах кожного чотирирічного циклу.

Клас **FCYCLE**

Призначення _ - формування лічильників циклу

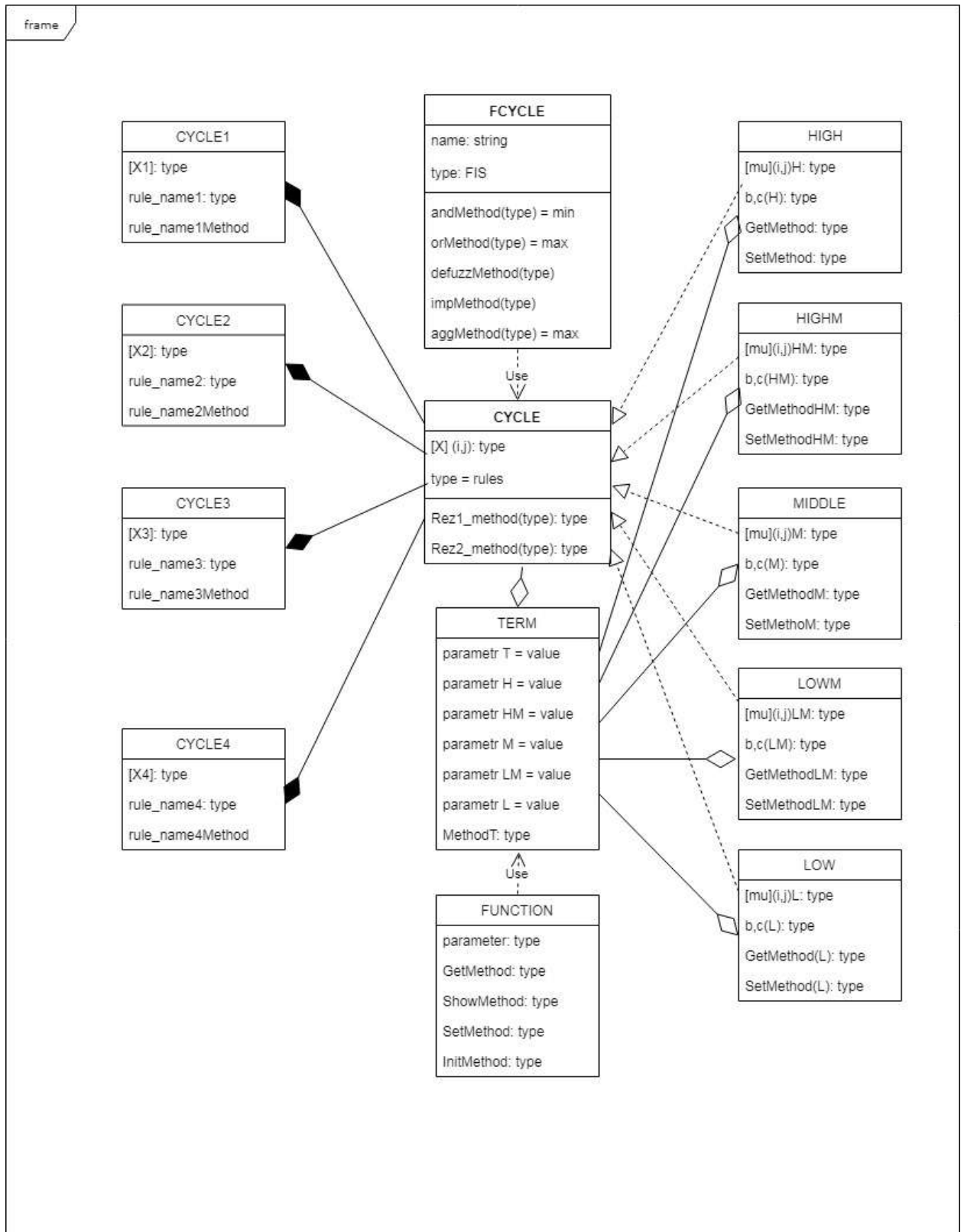


Рисунок 3.3. Діаграма класів модуля виведення частинного показника

3.4 Розробка алгоритма логічного виведення результату

Використовуючи апарат перетворень, який пропонується в [4], із співвідношень представлених в таб. 2.1 - 2.5 одержуємо нечіткі логічні рівняння, які після дефазифікації будуть мати вигляд (3.7) – (3.12)

$$\mu^{Q_j}(Y, Z, V) = \max_{p=1, h_j} \left\{ \min \left[\mu^{Y^{jp}}(Y), \mu^{Z^{jp}}(Z), \mu^{V^{jp}}(V) \right] \right\} \quad (3.1)$$

$$\mu^{Y_j}(y_1, y_2, \dots, y_5) = \max_{p=1, 5_j} \left\{ \min_{i=1, 10} \left[\mu^{y_i^{jp}}(y_i) \right] \right\} \quad (3.2)$$

$$\mu^{Z_j}(z_1, z_2, \dots, z_6) = \max_{p=1, 5_j} \left\{ \min_{i=1, 6} \left[\mu^{z_i^{jp}}(z_i) \right] \right\} \quad (3.3)$$

$$\mu^{V_j}(v_1, v_2, \dots, v_4) = \max_{p=1, 5_j} \left\{ \min_{i=1, 4} \left[\mu^{v_i^{jp}}(v_i) \right] \right\} \quad (3.4)$$

Таким чином нами отримана система співвідношень, яка повністю відповідає узагальненому дереву логічного висновку (рис. 2.6) і дозволяє обчислити ступені приналежності вектора значень вхідних змінних нечітким термам-оцінками вихідної змінної. Алгоритм нечіткого логічного висновку, що використовує узагальнене дерево виведення має вигляд:

1. Зафіксуємо вектор значень вхідних змінних

$$(y_1, y_2, \dots, y_{10}, z_1, z_2, \dots, z_5, v_1, v_2, \dots, v_4, w_1, w_2, x_1, x_2,)$$

2. Визначимо значення функцій приналежності термів-оцінок вхідних змінних.

3. Використовуючи співвідношення (3.1) - (3.4) обчислимо функції приналежності $\mu^{Q_j}(Y, Z, V)$ термів-оцінок вихідної величини Q , яка відповідає вектору значень вхідних змінних.

4. Визначимо оцінку Y_j , функція приналежності якої максимальна:

$$\mu^{Q_j}(Y, Z, V) = \max_{j=1,4} [\mu^{Q_j}(Y, Z, V)]$$

Відповідна схема алгоритму приведена на рис. 3.4.

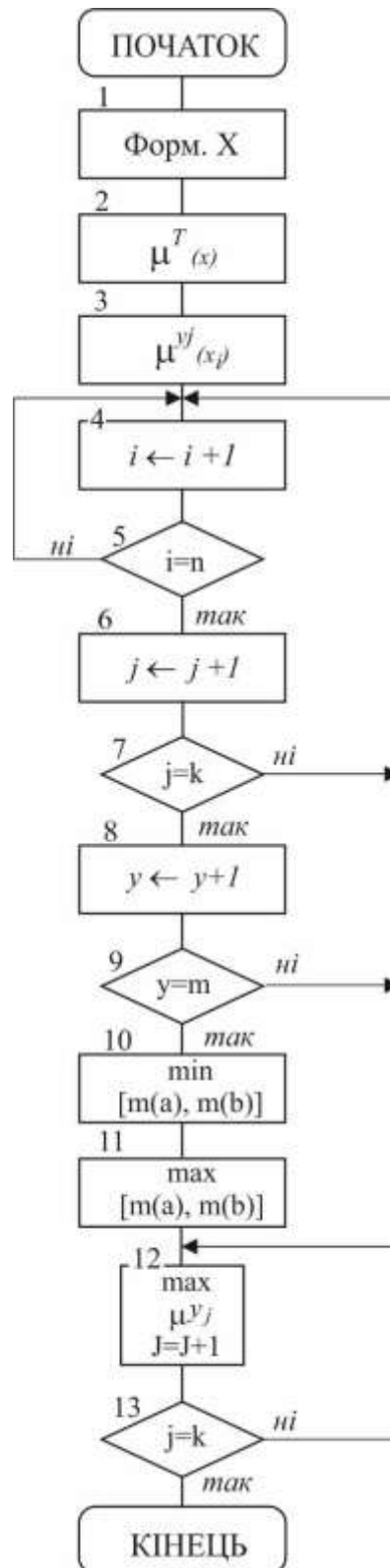


Рисунок 3.4 – Схема алгоритму логічного виведення рішення

3.5 Реалізація компонентів модуля з використанням екранних форм середовища FUZZY EXPERIENCE

Скористаємось екранною формою «Дерево виводу» середовища FUZZY EXPERIENCE і виконаємо програмну реалізацію цієї моделі.

Першим кроком буде визначення характеристик системи. Вводиться інформація про назву та призначення системи, визначається вихідна змінна.

Дерево логічного виводу формується шляхом послідовного виконання операцій додавання і / або видалення вузлів. При додаванні нового вузла запитується інформація про назву позначення (t), кількість термів для оцінки та їх назвах (<низька>, <нормальна>, <висока>). При додаванні вузла, відповідного входній змінній, запитується інформація про діапазон її зміни. Сформоване дерево логічного висновку зображено на рис. 3.5.

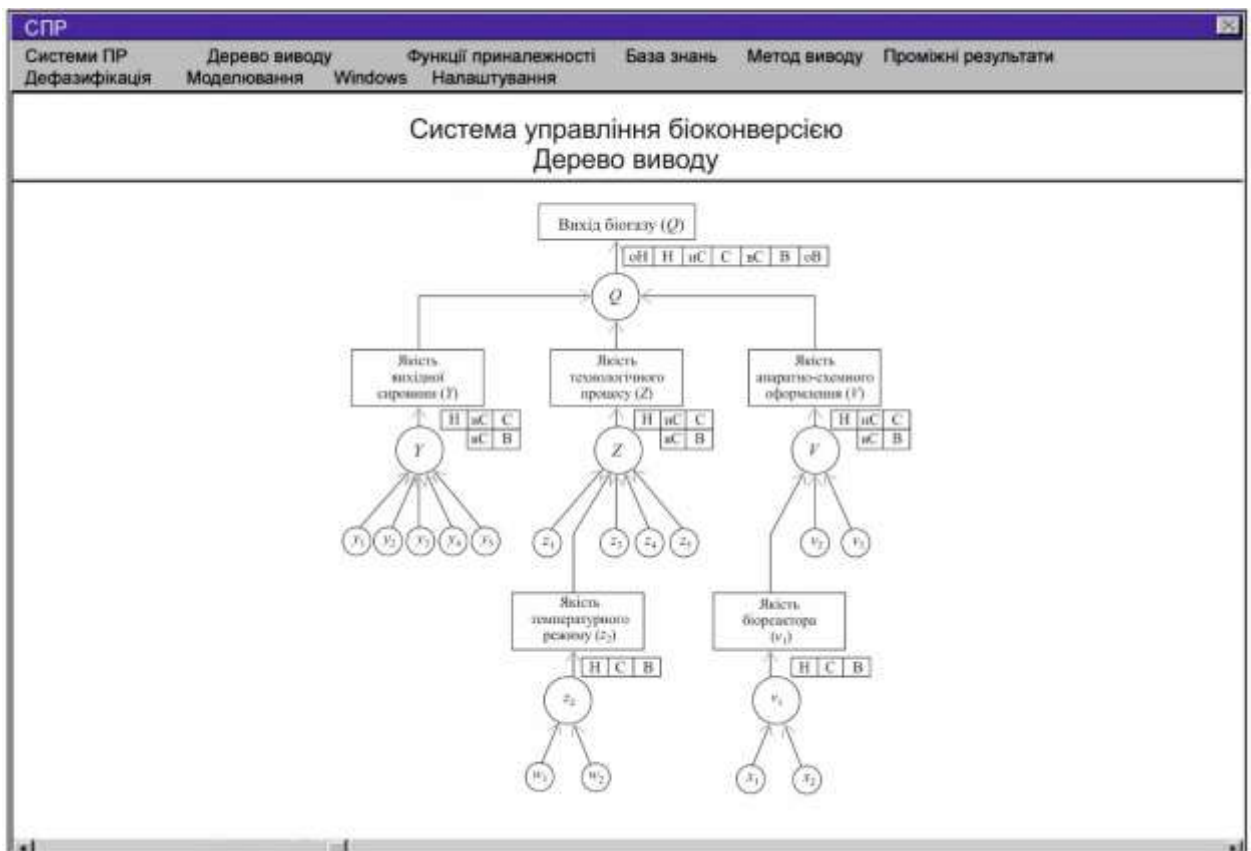


Рисунок 3.5 - Сформоване дерево логічного висновку

Визначимо функції приналежності лінгвістичних термів.

На цьому кроці визначаються моделі функцій належності, які використовуються для формалізації термів - оцінок змінних.

Визначимо експертні правила ЯКЦО-ТО, що описують поведінку об'єкта. Експертні правила ЯКЦО-ТО вносяться у відповідні матриці знань, рис.3.6. Налаштування нечіткої експертної системи шляхом вирішення задач оптимізації з використанням навчальної вибірки.

Введення значень вхідних змінних при виконанні розрахунків здійснюється у кількісній або якісній формі, а також по шкалі термометра (рис. 3.7).

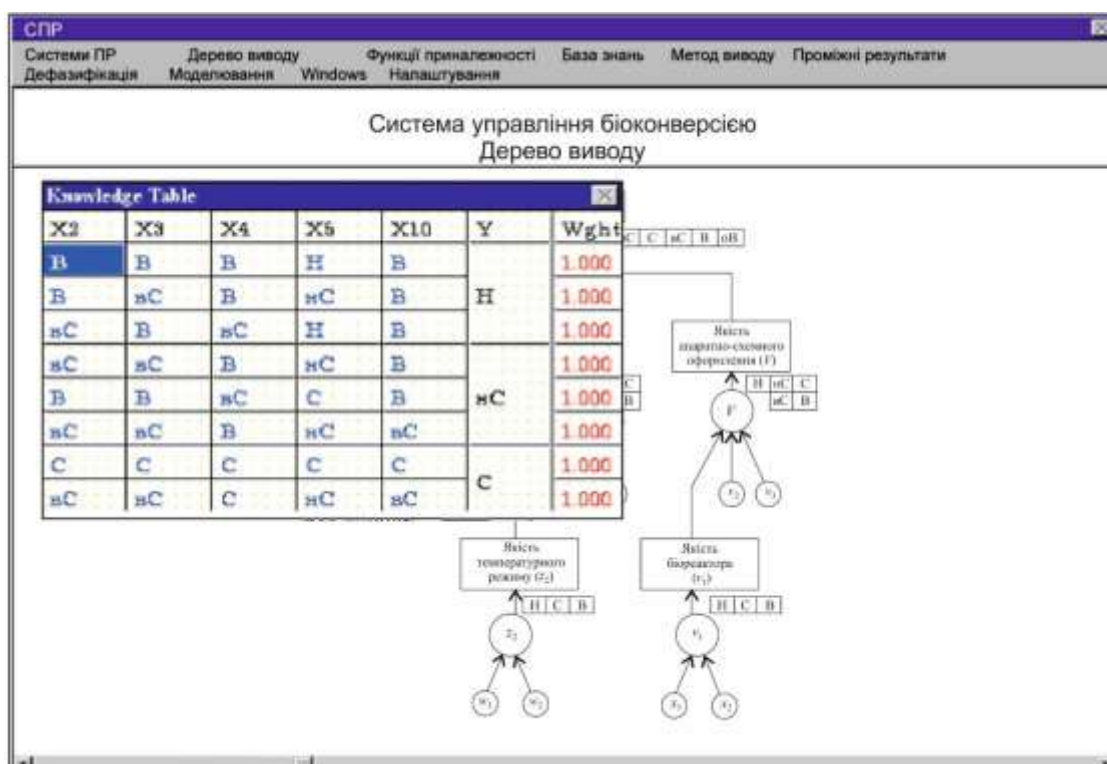


Рисунок 3.6 – Заповнення бази знань

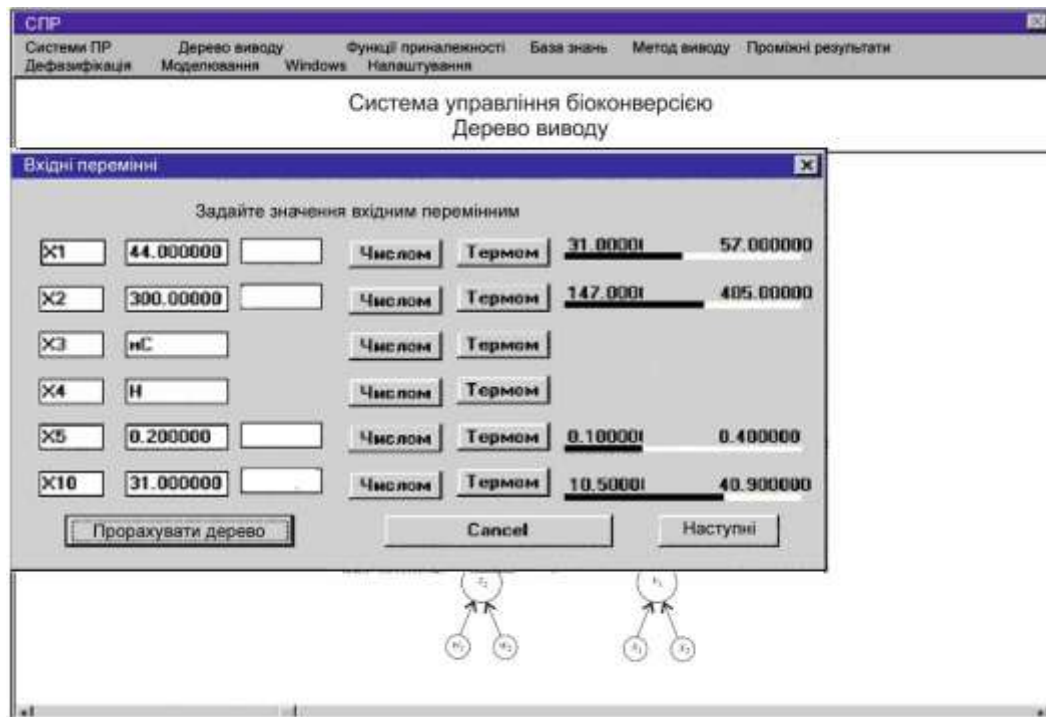


Рисунок 3.7 - Введення значень вхідних змінних

3.6 Висновок

В третьому розділі магістерської кваліфікаційної роботи виконане проектування та реалізація програмного модуля управління біоконверсією, що працює за доопрацьованою технологією.

В процесі створення програмного забезпечення виконано:

- обґрунтування та вибір середовища та мови програмування
- декомпозиція програмного забезпечення та розробка структури компонентів
- розробка діаграми класів
- розробка алгоритма логічного виведення результату
- реалізація компонентів модуля з використанням екранних форм середовища FUZZY EXPERIENCE.

Також проведено тестування розробленого модуля та розроблена інструкція користувача (ДОДАТОК А).

4 ЕКОНОМІЧНА ЧАСТИНА

4.1 Оцінювання комерційного потенціалу розробки

Метою проведення технологічного аудиту є оцінювання комерційного потенціалу розробки. Для проведення технологічного аудиту було залучено 2-х незалежних експертів. Такими експертами будуть доц. каф. КН Сілагін Олексій Віталійович та доц. каф. КН Арсенюк Ігор Ростиславович.

Здійснюємо оцінювання комерційного потенціалу розробки за 12-ма критеріями за 5-ти бальною шкалою.

Результати оцінювання комерційного потенціалу розробки наведено в таблиці 4.1.

Таблиця 4.1 – Результати оцінювання комерційного потенціалу розробки

Критерії	Прізвище, ініціали, посада експерта	
	1. Експерт 1	2. Експерт 2
	Бали, виставлені експертами:	
1	4	4
2	4	3
3	3	4
4	4	3
5	3	4
6	4	4
7	3	3
8	4	4
9	4	4
10	4	3
11	3	4
12	3	4
Сума балів	СБ ₁ = 44	СБ ₂ = 44

Продовження таблиці 4.1

Середньоарифметична сума балів \overline{CB}	$\overline{CB} = \frac{\sum_1^3 CB_i}{2} = 44$
---	--

Отже, з отриманих даних таблиці 4.1 видно, що нова розробка має високий рівень комерційного потенціалу.

4.2 Прогнозування витрат на виконання науково-дослідної роботи та конструкторсько–технологічної роботи

Для розробки нового програмного продукту необхідні такі витрати.

Основна заробітна плата для розробників визначається за формулою (4.1):

$$Z_o = \frac{M}{T_p} \cdot t, \quad (4.1)$$

де M- місячний посадовий оклад конкретного розробника;

T_p - кількість робочих днів у місяці, $T_p = 22$ дні;

t - число днів роботи розробника, t = 45 днів.

Розрахунки заробітних плат для керівника і програміста наведені в таблиці 4.2.

Таблиця 4.2 – Розрахунки основної заробітної плати

Працівник	Оклад M, грн.	Оплата за робочий день, грн.	Число днів роботи, t	Витрати на оплату праці, грн.
Науковий керівник	5500	250	6	1500
Інженер-програміст	4000	181,81	45	8181,45
Всього:				9681,45

Розрахуємо додаткову заробітну плату:

$$З_{\text{дод}} = 0,1 \cdot 9681,45 = 968,14 \text{ (грн.)}$$

Нарахування на заробітну плату операторів НЗП розраховується як 37,5...40% від суми їхньої основної та додаткової заробітної плати:

$$H_{\text{зп}} = (З_0 + З_p) \cdot \frac{\beta}{100}, \quad (4.2)$$

$$H_{\text{зп}} = (9681,45 + 968,14) \cdot \frac{36,3}{100} = 3865,80 \text{ (грн.)}.$$

Розрахунок амортизаційних витрат для програмного забезпечення виконується за такою формулою:

$$A = \frac{Ц \cdot H_a}{100} \cdot \frac{T}{12}, \quad (4.3)$$

де Ц – балансова вартість обладнання, грн;

H_a – річна норма амортизаційних відрахувань % (для програмного забезпечення 25%);

T – Термін використання (T=3 міс.).

Таблиця 4.3 – Розрахунок амортизаційних відрахувань

Найменування програмного забезпечення	Балансова вартість, грн.	Норма амортизації, %	Термін використання, міс.	Величина амортизаційних відрахувань, грн
Персональний комп'ютер	10000	25	3	625
Всього:				625

Розрахуємо витрати на комплектуючі. Витрати на комплектуючі розрахуємо за формулою:

$$K = \sum_1^n H_i \cdot C_i \cdot K_i, \quad (4.4)$$

де n – кількість комплектуючих;

H_i - кількість комплектуючих i -го виду;

C_i – покупна ціна комплектуючих i -го виду, грн;

K_i – коефіцієнт транспортних витрат (приймемо $K_i = 1,1$).

Таблиця 4.4 - Витрати на комплектуючі, що були використані для розробки ПЗ.

Найменування матеріалу	Одиниці виміру	Ціна, грн.	Витрачено	Вартість витрачених матеріалів, грн.
Флешка	шт.	180	1	180
Пачка паперу	уп.	115	1	115
Ручка	шт.	5	1	5
Всього з урахуванням транспортних витрат				330

Витрати на силову електроенергію розраховуються за формулою:

$$V_e = V \cdot P \cdot \Phi \cdot K_{\Pi} ; \quad (4.5)$$

де V – вартість 1кВт-години електроенергії ($V=1,7$ грн/кВт);

P – установлена потужність комп'ютера ($P=0,6$ кВт);

Φ – фактична кількість годин роботи комп'ютера ($\Phi=200$ год.);

K_{Π} – коефіцієнт використання потужності ($K_{\Pi} < 1$, $K_{\Pi} = 0,7$).

$$V_e = 1,7 \cdot 0,6 \cdot 200 \cdot 0,7 = 142,8 \text{ (грн.)}$$

Розрахуємо інші витрати $V_{ін}$.

Інші витрати I_b можна прийняти як (100...300)% від суми основної заробітної плати розробників та робітників, які були виконували дану роботу, тобто:

$$V_{ін} = (1..3) \cdot (Z_o + Z_p). \quad (4.6)$$

Отже, розрахуємо інші витрати:

$$V_{ін} = 1 \cdot (9681,45 + 968,14) = 10649,59 \text{ (грн.)}$$

Сума всіх попередніх статей витрат дає витрати на виконання даної частини роботи:

$$V = Z_o + Z_d + H_{зп} + A + K + V_e + I_b$$

$$V = 9681,45 + 968,14 + 3865,80 + 625 + 330 + 142,8 + 10649,59 = 26262,78 \text{ (грн.)}$$

Розрахуємо загальну вартість наукової роботи $B_{заг}$ за формулою:

$$B_{заг} = \frac{V_{ін}}{\alpha} \quad (4.7)$$

де

α – частка витрат, які безпосередньо здійснює виконавець даного етапу роботи, у відн. одиницях = 1.

$$B_{заг} = \frac{26262,78}{1} = 26262,78$$

Прогнозування загальних витрат ЗВ на виконання та впровадження результатів виконаної наукової роботи здійснюється за формулою:

$$ЗВ = \frac{B_{заг}}{\beta} \quad (4.8)$$

де

β – коефіцієнт, який характеризує етап (стадію) виконання даної роботи.

Отже, розрахуємо загальні витрати:

$$ЗВ = \frac{26262,78}{0,9} = 29180,86 \text{ (грн.)}$$

4.3 Прогнозування комерційних ефектів від реалізації результатів розробки

Спрогнозуємо отримання прибутку від реалізації результатів нашої розробки. Зростання чистого прибутку можна оцінити у теперішній вартості грошей. Це забезпечить підприємству (організації) надходження додаткових коштів, які дозволять покращити фінансові результати діяльності .

Оцінка зростання чистого прибутку підприємства від впровадження результатів наукової розробки. У цьому випадку збільшення чистого прибутку підприємства $\Delta \Pi_i$ для кожного із років, протягом яких очікується отримання позитивних результатів від впровадження розробки, розраховується за формулою:

$$\Delta \Pi_i = \sum_1^n (\Delta \Pi_{я} \cdot N + \Pi_{я} \Delta N)_i \quad (4.9)$$

де

$\Delta \Pi_{я}$ – покращення основного якісного показника від впровадження результатів розробки у даному році;

N – основний кількісний показник, який визначає діяльність підприємства у даному році до впровадження результатів наукової розробки;

ΔN – покращення основного кількісного показника діяльності підприємства від впровадження результатів розробки;

$\Pi_{\text{я}}$ – основний якісний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки;

n – кількість років, протягом яких очікується отримання позитивних результатів від впровадження розробки.

В результаті впровадження результатів наукової розробки витрати на виготовлення інформаційної технології зменшаться на 20 грн (що автоматично спричинить збільшення чистого прибутку підприємства на 20 грн), а кількість користувачів, які будуть користуватись збільшиться: протягом першого року – на 200 користувачів, протягом другого року – на 175 користувачів, протягом третього року – 125 користувачів. Реалізація інформаційної технології до впровадження результатів наукової розробки складала 600 користувачів, а прибуток, що отримував розробник до впровадження результатів наукової розробки – 300 грн.

Спрогнозуємо збільшення чистого прибутку від впровадження результатів наукової розробки у кожному році відносно базового.

Отже, збільшення чистого продукту $\Delta \Pi_1$ протягом першого року складатиме:

$$\Delta \Pi_1 = 20 \cdot 600 + (300 + 20) \cdot 200 = 76000 \text{ грн.}$$

Протягом другого року:

$$\Delta \Pi_2 = 20 \cdot 600 + (300 + 20) \cdot (200 + 175) = 132000 \text{ грн.}$$

Протягом третього року:

$$\Delta\Pi_3 = 20 \cdot 600 + (300 + 20) \cdot (200 + 175 + 125) = 172000 \text{ грн.}$$

4.4 Розрахунок ефективності вкладених інвестицій та період їх окупності

Визначимо абсолютну і відносну ефективність вкладених інвестором інвестицій та розрахуємо термін окупності.

Абсолютна ефективність $E_{\text{абс}}$ вкладених інвестицій розраховується за формулою:

$$E_{\text{абс}} = (\text{ПП} - PV), \quad (4.10)$$

де $\Delta\Pi_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої НДДКР, грн;

t – період часу, протягом якого виявляються результати впровадженої НДДКР, 3 роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні; для України цей показник знаходиться на рівні 0,1;

t – період часу (в роках) від моменту отримання чистого прибутку до точки 2, 3, 4.

Рисунок, що характеризує рух платежів (інвестицій та додаткових прибутків) буде мати вигляд, рисунок 4.1.

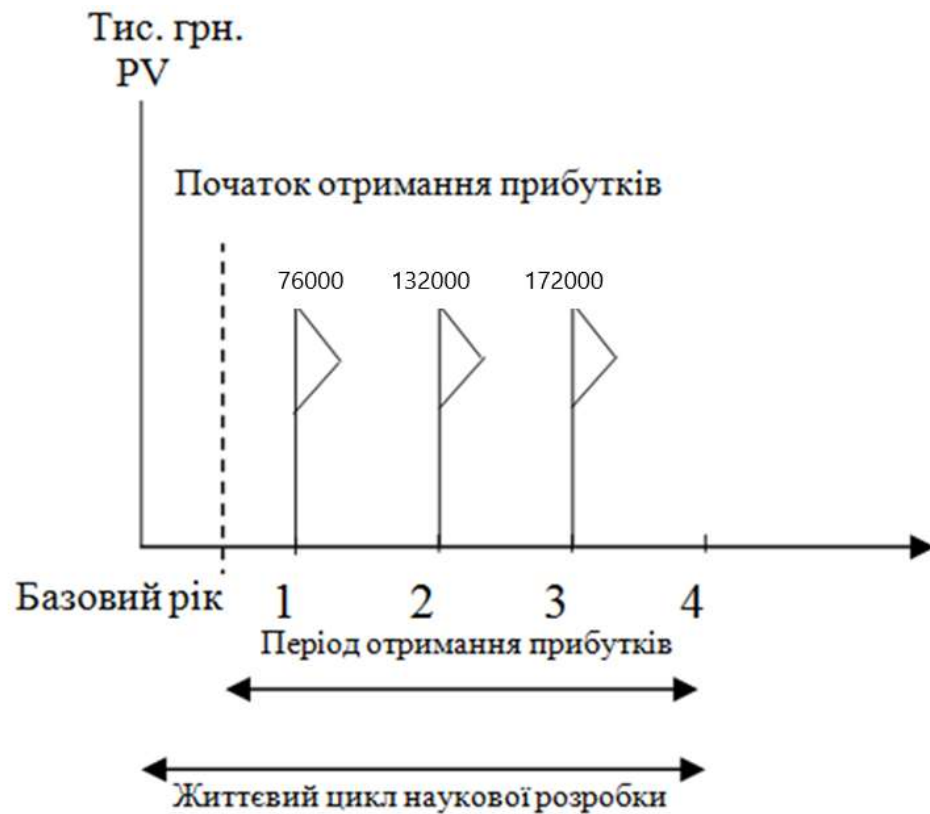


Рисунок 4.1 – Вісь часу з фіксацією платежів, що мають місце під час розробки та впровадження результатів НДДКР

Розрахуємо вартість чистих прибутків за формулою:

$$\text{ПП} = \sum_1^m \frac{\Delta\Pi_i}{(1+\tau)^t} \quad (4.11)$$

де $\Delta\Pi_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої НДДКР, грн;

t – період часу, протягом якого виявляються результати впровадженої НДДКР, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні; для України цей показник знаходиться на рівні 0,1;

t – період часу (в роках) від моменту отримання чистого прибутку до точки.

Отже, розрахуємо вартість чистого прибутку:

$$\text{ПП} = \frac{29180,86}{(1+0,1)^0} + \frac{76000}{(1+0,1)^2} + \frac{132000}{(1+0,1)^3} + \frac{172000}{(1+0,1)^4} = 308642,63 \text{ (грн.)}$$

Тоді розрахуємо $E_{\text{абс}}$:

$$E_{\text{абс}} = 308642,63 - 29180,86 = 279461,77 \text{ грн.}$$

Оскільки $E_{\text{абс}} > 0$, то вкладання коштів на виконання та впровадження результатів НДДКР буде доцільним.

Розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій $E_{\text{в}}$ за формулою:

$$E_{\text{в}} = \sqrt[T]{1 + \frac{E_{\text{абс}}}{\text{PV}}} - 1 \quad (4.12)$$

де

$E_{\text{абс}}$ – абсолютна ефективність вкладених інвестицій, грн;

PV - теперішня вартість інвестицій $\text{PV} = \text{ЗВ}$, грн;

$T_{\text{ж}}$ – життєвий цикл наукової розробки, роки.

Тоді будемо мати:

$$E_{\text{в}} = \sqrt[3]{1 + \frac{279466,71}{29180,86}} - 1 = 1,19 \text{ або } 119 \%$$

Далі, розраховану величина E_b порівнюємо з мінімальною (бар'єрною) ставкою дисконтування $\tau_{\text{мін}}$, яка визначає ту мінімальну дохідність, нижче за яку інвестиції вкладатися не будуть. У загальному вигляді мінімальна (бар'єрна) ставка дисконтування $\tau_{\text{мін}}$ визначається за формулою:

$$\tau = d + f,$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2019 році в Україні $d = 0,2$;

f – показник, що характеризує ризикованість вкладень, величина $f = 0,1$.

$$\tau = 0,2 + 0,1 = 0,3$$

Оскільки $E_b = 119\% > \tau_{\text{мін}} = 0,3 = 30\%$, то у інвестор буде зацікавлений вкладати гроші в дану наукову розробку.

Термін окупності вкладених у реалізацію наукового проекту інвестицій. Термін окупності вкладених у реалізацію наукового проекту інвестицій $T_{\text{ок}}$ розраховується за формулою:

$$T_{\text{ок}} = \frac{1}{E_b}$$

$$T_{\text{ок}} = \frac{1}{1,19} = 0,84 \text{ року}$$

Обрахувавши термін окупності даної наукової розробки, можна зробити висновок, що фінансування даної наукової розробки буде доцільним.

4.5 Висновок

В четвертому розділі магістерської кваліфікаційної роботи були виконані наступні дослідження економічного характеру:

- оцінювання комерційного потенціалу розробки;
- прогнозування витрат на виконання науково-дослідної роботи та конструкторсько-технологічної роботи;
- прогнозування комерційних ефектів від реалізації результатів розробки;
- розрахунок ефективності вкладених інвестицій та період їх окупності.

В результаті проведення цих досліджень були одержані наступні показники економічної ефективності цієї розробки.

Щорічна ефективність вкладених в наукову розробку інвестицій складає 119%, що набагато вище за мінімальну бар'єрну ставку дисконтування, яка складає 25%. Це означає потенційну зацікавленість інвесторів у фінансуванні розробки.

Термін окупності вкладених у реалізацію проекту інвестицій становить 0,84 року, що також свідчить про доцільність фінансування нової розробки.

ВИСНОВКИ

Всі завдання, поставлені в магістерській кваліфікаційній роботі, виконано в повному об'ємі, а саме:

- обґрунтована доцільність доопрацювання інформаційної технології управління біоконверсією;
- проаналізовані існуючі технології, методи і моделі управління біоконверсією;
- сформульовані вимоги до роботи технології та розроблено ТЗ;
- доопрацьована існуюча інформаційна технологія управління біоконверсією з врахуванням додаткових факторів, таких як стабільність температурного режиму та якість додаткового обладнання;
- проведено математичне моделювання роботи етапів доопрацьованої технології з використанням апарату нечіткої логіки;
- проведено фазифікацію розробленої моделі;
- розроблено та наповнено базу знань у вигляді матриць з правилами ЯКЩО-ТО;
- на основі доопрацьованої технології виконано проектування програмного модуля управління біоконверсією;
- реалізовано та налаштовано роботу програмного модуля управління біоконверсією;
- протестовано роботу налаштованого програмного модуля;
- виконано задачі економічного розділу.

Щорічна ефективність вкладених в наукову розробку інвестицій складає 119%, що набагато вище за мінімальну бар'єрну ставку дисконтування, яка складає 25%. Це означає потенційну зацікавленість інвесторів у фінансуванні розробки.

Термін окупності вкладених у реалізацію проекту інвестицій становить 0,84 року, що також свідчить про доцільність фінансування нової розробки.

Мета дослідження - підвищення рівня виходу біогазу в процесі біоконверсії досягається за рахунок підвищення адекватності моделі, в якій враховані додаткові фактори, такі як, стабільність температурного режиму та якість додаткового обладнання. За твердженнями експертів управління цими факторами підвищує вихід біогазу на 3-4% [4]

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1 Когут В.В, Поліщук Б.В, Сілагін О.В, Шалак А.В. Інжинірингова технологія створення експертних систем нечіткої логіки в *Матеріалах конференції «Молодь в науці: дослідження, проблеми, перспективи (МН-2020)»*, Вінниця, 2020. [Електронний ресурс].
Режимдоступу: <https://conferences.vntu.edu.ua/index.php/all-mn/index/pages/view/zbirn2020> Дата звернення: Черв. 2020
- 2 Шлейкин А.Г., Жилинская Н.Т. Введение в биотехнологию. – СПб., 2013. – 95 с. 2.
- 3 Сидоренко О.Д., Кутровский В.Н. Биоконверсия отходов агропромышленного комплекса. – М.: НИЦ Инфра-М, 2013. – 160 с.
- 4 Инженерная энзимология / В.К. Османов, О.В. Бирюкова, А.В. Борисов, Г.Н. Борисова, Ж.В. Мацулевич: Учеб. пособие. – Нижний Новгород: Изд-во Нижегородской гос. мед. акад., 2005. – 75 с.
- 5 Ресурси ІНТЕРНЕТ www.wikipedia.org.
- 6 Ротштейн О.П. Інтелектуальні технології ідентифікації: нечіткі множини, генетичні алгоритми, нейронні мережі. – Вінниця: Універсам – Вінниця, 1999. – 320с., іл..
- 7 Г. Джойс. Языки систем искусственного интеллекта.—М.: Мир, 1994.
- 8 Искусственный интеллект. Справочник в 3-х томах. - М.: Радио и связь, 1990.
- 9 Буч Г., Якобсон А., Рамбо Дж. UML. Классика CS. 2-е изд./ Пер. с англ.: Под общей редакцией проф. С. Орлова- СПб.: Питер, 2006.-736 с.:ил.
- 10 Общая алгебра. Т. 2/ В. Артамонов, В.Н.Салий, Л.А.Скорняков и др. Под общ. Ред. Л.А.Скорнякова.-М.: Наука. Гл. ред. Физ.-мат. Лит., 1991. –480 с.

- 11 Вагин В.Н. Дедукция и обобщение в системах принятия решений. – М.: Наука, Гл. ред. физ.-мат. лит.; 1988, - 384 с.
- 12 Буч Г., Рамбо Д., Якобсон А. Язык UML. Руководство пользователя: Пер. с англ. – М.: ДМК, 2000. 432 с.: ил.
- 13 Кватрани Т. Rational Rose 2000 и UML. Визуальное моделирование: Пер. с англ. – М.: ДМК Пресс, 2001. – 176 с.: ил.
- 14 Ларман К. Применение UML и шаблонов проектирования.: Пер. с англ. : Уч. Пос. – М.: Издательский дом «Вильямс», 2001. – 496 с.: ил.
- 15 Трофимов С.А. Case- технологии: практическая работа в Rational Rose – М.: ЗАО «Издательство БИНОМ», 2001 г. – 272 с.: ил.
- 16 www.biotechnolog.ru;
- 17 www.rusbiotech.ru;
- 18 www.fermenter.ru;
- 19 www.inbio.ru;
- 20 www.microzym.ru.
- 21 А. Новичков. Эффективная разработка программного обеспечения с использованием технологий и инструментов компании RATIONAL.
www.interface.com
- 22 Case Studies in Object Oriented Analysis and Design. E. Yourdon, C. Argila. Copyright 1996 by Prentice-Hall PTR. Prentice-Hall, Inc. A Simon & Schuster Company. Upper Saddle River, New Jersey 07458, p.2642.
Ломоносов, М. В. Полн. собр. соч. – М. ; Л. : Изд-во АН СССР. 1950. – Т. 1. – 620 с.