Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)
Кафедра Штучного інтелекту
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Дослідження особливостей побудови інтелектуальних експертних систем для прогнозування та діагностики несправностей
(тема)

Виконав:
студент 2 курсу, групи СШІм-19-2
Степаненко Є.В.
(прізвище, ініціали)

Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)
Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)
Освітня програма Системи штучного інтелекту
(повна назва спеціалізації)

Керівник проф. Філатов В.О
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ В.О. Філатов
(підпис) (прізвище, ініціали)

2021 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Штучного інтелекту
(повна назва)

Рівень вищої освіти другий (магістерський)

Спеціальність 122 Комп'ютерні науки
(код і повна назва)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма Системи штучного інтелекту (СШІ)
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«___» _____ 20 ___ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Ступаненку Євгену Віталійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження особливостей побудови інтелектуальних експертних систем для прогнозування та діагностики несправностей

затверджена наказом університету від 29 березня 20 21 р. № 390СТ

2. Термін подання студентом роботи до екзаменаційної комісії ___ _____ 20 ___ р.

3. Вихідні дані до роботи Функція: Огляд сучасного стану використання експертних систем методів прогнозування та діагностики несправностей. Розробка прототипу експертної системи для оптимізації роботи диспетчера електророзподільної станції. Перелік використовуваних програмних засобів: ОС Windows 10, середовище розробки PhpStorm міжплатформленне середовище розробки Unity 3D, СУБД PostreSQL.

Технічне забезпечення: Різні види ЕОМ: ПК, смартфон, планшет, ноутбук зі встановленим програмним середовищем

4. Перелік питань, що потрібно опрацювати в роботі Аналіз предметної області; Постанова задач досліджень; Характеристика предметної області; Проблематика предметної області; Аналіз методів побудови рекомендацій, заснованих на знаннях; Розробка прототипу експертної системи; Огляд використаного середовища та набору даних; Передобробка даних; Реалізація методів класифікації; Демонстрація роботи системи. Тестування системи

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Рисунок 1.1 – Класифікація інтелектуальних інформаційних систем; Рисунок 1.2 – Етапи розробки експертної системи; Рисунок 1.3 – Типова структура експертної системи; Рисунок 3.1 – Класифікація моделей подання знань; Рисунок 3.2 – Приклад продукційної моделі; Рисунок 3.3 – Приклад семантичної мережі; Рисунок 3.4 – Структура фрейма; Рисунок 3.5 – Приклад фреймової мережі; Рисунок 3.6 – Інтерфейс ПО Unity 3D; Рисунок 3.7 – Інтерфейс PhpStorm; Рисунок 3.8 – Розподілена електрична мережа України; Рисунок 3.9 – Алгоритм Дейкстри на графі; Рисунок 3.10 – Формальне представлення алгоритму Дейкстри; Рисунок 3.11 – Гамільтонів шлях на графі; Рисунок 3.12 – Формальне представлення алгоритму проходження Гамільтонового шляху; Рисунок 3.13 – Дерево шляхів з вершини 1; Рисунок 4.1 – Початковий стан системи; Рисунок 4.2 – Стан системи при аварії; Рисунок 4.3 – Встановлення всіх пов'язаних підстанцій в стан «Аварії»; Рисунок 4.4 – Встановлення ваг і обнуління id вузлів; Рисунок 4.5 – Заповнення графа за допомогою розрахунку вузлів; Рисунок 4.6 – Заповнення графа щодо резервних ребер; Рисунок 4.7 – Знайдений маршрут; Рисунок 4.8 – Таблиця БД «Edges»; Рисунок 4.9 – Таблиця БД Vertices; Рисунок 4.10 – Зміст таблиці «Edge_status»; Рис. 4.11 – Початковий стан графа; Рисунок 4.12 – Результат роботи ЕС при пошкодженні 4-5 фідера; Рисунок 4.13 – Робота ЕС при двох аваріях; Рисунок 4.14 – Зміни в БД

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на атестаційне проектування	29.03.2021	виконано
2	Аналіз завдання, пошук літератури та аналогів з теми	30.03.2021	виконано
3	Опрацювання літератури та аналіз об'єкту	30.03.2021	виконано
4	Вибір оптимального підходу до розробки	01.04.2021	виконано
5	Проектування програмного засобу	01.04.2021 –15.04.2021	виконано
6	Вибір технічних засобів для розробки ПЗ	15.04.2021	виконано
7	Розробка програмного засобу	15.04.2021 – 20.04.2021	виконано
8	Аналіз результатів отриманих за допомогою ПЗ	21.04.2021	виконано
9	Оформлення пояснювальної записки та документації	21.04.2021 –22.04.2021	виконано
10	Оформлення графічної та презентаційної частини	22.04.2021	виконано
11	Попередній захист	13.05.2021	виконано
12	Захист перед ЕК	19.05.2021	

Дата видачі завдання 29 березня 20 21 р.

Студент _____
(підпис)

Керівник роботи _____ _____
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 95 с., 31 рис., 3 дод., 28 джерел.

ЕКСПЕРТНА СИСТЕМА, ІНЖЕНЕРІЯ ЗНАНЬ, ІНТЕЛЕКТУАЛЬНА ІНФОРМАЦІЙНА СИСТЕМА, ПОКРАЩЕННИЙ АЛГОРИТМ ДЕЙКСТРИ, ПРОГРАМНИЙ ДОДАТОК, PHP, SQL, UNITY.

Об’єкт розробки – експертна система діагностики електричної мережі та автоматизації роботи диспетчера електророзподільної станції.

Предмет розробки – обробка та аналіз даних, щодо зміни стану роботи об’єктів електричної мережі за допомогою експертної системи.

Мета атестаційної роботи – розробка експертної системи, як елементу програмного додатку, завданням якого є обробка та аналіз даних, щодо зміни стану роботи підстанцій в електричній мережі.

Методи розробки – створення програмних додатків на мові програмування PHP, методи розробки експертних систем, методи побудови продукційної моделі, алгоритми обходу графа.

Результати кваліфікаційної роботи – програмний додаток, розрахований на міжплатформленне користування, обробки та аналізу даних електророзподільної станції.

Область застосування – програмний продукт використовується для обробки даних електричної мережі та подальшого їх аналізу для автоматизації роботи диспетчера електророзподільної підстанції, для забезпечення живленням всіх користувачів мережі, запобігання НС.

РЕФЕРАТ

Пояснительная записка: 95 с., 31 рис., 3 прил., 28 источников.

ИНЖЕНЕРИЯ ЗНАНИЙ, ИНТЕЛЛЕКТУАЛЬНАЯ ИНФОРМАЦИОННАЯ СИСТЕМА, ПРОГРАММНОЕ ПРИЛОЖЕНИЕ, УЛУЧШЕНН АЛГОРИТМ ДЕЙКСТРЫ, ЭКСПЕРТНАЯ СИСТЕМА, PHP, UNITY, SQL.

Объект разработки – экспертная система диагностики электрической сети и автоматизации работы диспетчера электрораспределительной станции.

Предмет разработки – обработка и анализ данных по изменению состояния работы объектов электрической сети с помощью экспертной системы.

Цель аттестационной работы – разработка экспертной системы как элемента программного приложения, задачей которого является обработка и анализ данных по изменению состояния работы подстанций в электрической сети.

Методы разработки – создание приложений на языке программирования PHP, методы разработки экспертных систем, методы построения продукционной модели, алгоритмы обхода графа.

Результат квалификационной работы – программный продукт, рассчитанный на межплатформенное использование, обработки и анализа данных электрораспределительной станции.

Область применения – программный продукт используется для обработки данных электрической сети и дальнейшего их анализа для автоматизации работы диспетчера электрораспределительной подстанции для обеспечения питанием всех пользователей сети, предотвращения ЧС.

ABSTRACT

Explanatory note: 95 p., 31 fig., 3 ann., 28 sources.

DICKSTER ENHANCED ALGORITHM, EXPERT SYSTEM, INTELLECTUAL INFORMATION SYSTEM, KNOWLEDGE ENGINEERING, PHP, SOFTWARE APPLICATION, SQL, UNITY.

The object of development is an expert system of diagnostics of the electric network and automation of the work of the dispatcher of the power distribution station.

The subject of development is the processing and analysis of data on changes in the state of operation of electrical network facilities using an expert system.

The purpose of the certification work is to develop an expert system as an element of a software application, the task of which is data processing and analysis, to change the state of operation of substations in the electrical network.

Development methods – creation of software applications in PHP programming language, methods of development of expert systems, methods of construction of production model, algorithms of bypassing the graph.

The results of the qualification work - a software application designed for cross-platform use, processing and analysis of data from the power distribution station.

Scope – the software product is used for data processing of the electric network and their further analysis for automation of work of the dispatcher of electric distribution substation, for providing power to all users of a network, prevention of emergency situations.

ЗМІСТ

Вступ 11
1 Аналіз сучасного стану експертних систем в області діагностики 15
1.1 Поняття інтелектуальної інформаційної системи 15
1.2 Відмінності експертних систем від інших видів програм штучного інтелекту 16
1.2.1 Системи з інтелектуальним інтерфейсом 16
1.2.2 Експертні системи 17
1.2.3 Самонавчальні системи 18
1.2.4 Адаптивні інформаційні системи 19
1.3 Особливості експертних систем 20
1.4 Переваги використання експертних систем 21
1.5 Області використання ЕС 22
1.6 Функції експертної системи 26
1.7 Структура та режими використання ЕС 28
1.8 Класифікація ЕС 30
1.9 Діагностика знань та умінь за допомогою експертних систем 30
1.10 Обмеження в застосування експертних систем 33
2 Використання експертної системи в електроенергетиці 35
2.1 Роль експертних систем в електроенергетиці 35
2.2 Проведення процедури прийняття рішення 36
2.3 Формалізація логіки оперативних перемикань в електричних мережах 37
2.4 Опис основних методів оптимізації режимів напруги в розподільній мережі 39
3 Проектування моделі системи 41
3.1 Бази знань та моделі представлення знань 41
3.1.1 База знань 41
3.1.2 Модель представлення знань 43
3.1.2.1 Продукційна модель 44

3.1.2.2 Семантичні мережі ..46
3.1.2.3. Фрейми ..48
3.1.2.4 Формальні логічні моделі 51
3.2 Аналіз та вибір системи управління базою даних експертної системи діагностики 52
3.3 Вибір програмного забезпечення 55
3.3.1 Unity 3D 56
3.3.2 IDE PhpStorm 57
3.4 Вибір алгоритму для обходу графа 59
3.4.1 Алгоритм Дейкстри 61
3.4.2 Гамільтонів шлях ..65
3.4.3 Покращений алгоритм Дейкстри 67
4 Розробка алгоритму та його реалізація 69
4.1 Функціональний аналіз мережі 69
4.2 Розробка алгоритму та принцип його роботи 70
4.3 Використовуваний стек технологій 74
4.4 Опис експертної системи 76
4.5 Тестування експертної системи 79
Висновки 82
Перелік джерел посилання 84
Додаток А Програмний код використаного алгоритму 87
Додаток Б Програмний код контролерів 91
Додаток В Відомість кваліфікаційної роботи магістра 92

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

АРМ – автоматизоване робоче місце;

БД – база даних;

ЕОМ – електронно-обчислювальна машина;

ЕС – експертна система;

ІІС – інтелектуальна інформаційна система;

ПБЗ – перемикання без збудження;

ПК – персональний комп’ютер;

ПМ-інтерфейс – природно-мовний інтерфейс;

РП – робоча пам’ять;

РПН – регулювання напруги під навантаженням;

СУБД – система управління баз даних;

ТЕЦ – теплоелектроцентраль;

АКО – A-Kind-Of – вид;

MVCC – multiversion concurrency control – управління паралельним доступом за допомогою багатоверсійності;

SQL – structured query language – декларативна мова програмування;

UI – user interface – інтерфейс користувача.

ВСТУП

Складно уявити сучасний світ без електрики. Електрику проводять в найвіддаленіші місця нашої країни, для здійснення виробничої, промислової або побутової діяльності. Електричні мережі України мають протяжність понад 930 тисяч кілометрів, а трансформаторна потужність підстанцій та трансформаторних пунктів понад 78 тис МВА.

Очевидно, що відсутність електроенергії може стати великою проблемою для населення та катастрофою на підприємствах, які використовують легкозаймисті рідини або мають виробництво з наслідками появи вугільного пилу, деревної муки. Найчастіше, на підстанції з напругою від 6 до 10 кВ припадає 70% відмов електричної мережі, що говорить про великі витрати і часте відключення споживачів від мережі.

Більшою мірою, показником надійності електричної мережі є термін служби трансформаторів і електроустаткування. Існуючі розподільні мережі спроектовані за радіальним принципом і, найчастіше, складаються з слабо надійних елементів. У більшості мереж використовуються алюмінієві дроти з маленькою площею поперечного перерізу, мала міцність опор. Природні лиха також є причинами відмов. За критеріями надійності електропостачання та якості електроенергії, на даний момент, розподільні мережі, не відповідають вимогам, що пред'являються до електричних підстанцій 6 та 10 кВ. Більш того, на них практично відсутнє обладнання для автоматизації діагностики та попередження аварій.

Для того, щоб підвищити термін служби обладнання в 1.5 рази, необхідно проводити капітальний ремонт електрообладнання, застосовувати нові системи автоматизації, які можуть дозволяти домагатися більш якісного обслуговування і управління аваріями в розподільних електричних мережах.

Управляє розподільними електричними мережами – диспетчер. Інформація, якою оперує диспетчерський персонал, має бути максимально

надійною і повною, задля підвищення якості прийнятих рішень. Диспетчер обробляє величезний трафік інформації, проте в даний час, коли стрімко розвиваються комп'ютерні обчислювальні технології, з'явилася можливість істотно змінити підхід до виявлення аварій, прискорити процес їх усунення.

З'явилося абсолютно нове рішення, яке носить назву «експертна система». Дана система допомагає диспетчеру зменшити кількість інформації, яку необхідно тримати в голові, а також виступає в якості повноцінного порадника чи експерта.

Якщо передати більшу частину функцій, які виконуються диспетчером до експертної системи, це дозволить зменшити час прийняття рішень, а також звільнить диспетчерів від щоденної рутинної роботи, і що на найбільш важливо – це допоможе зменшити кількість помилок в разі виникнення аварійних ситуацій.

На сьогоднішній день, все більше і більше підприємств вкрай залежні від електрики, настільки, що довге відключення електроенергії може викликати НС на виробництві. Через те, що диспетчеру доводиться обробляти величезну кількість інформації, на етапі вибору альтернативного маршруту виникають проблеми з географічним розташуванням вузлів. Тому постають питання про: економічні витрати, необхідність врахування людських ресурсів, навантаження на підстанціях, пріоритетність підстанцій, необхідність перевірки відсутності «подвійного» включення, складення заявки для групи швидкого реагування і так далі.

Помилки в роботі оператора можуть привести до ще більших згубних наслідків, ніж відсутність електроенергії у споживачів.

Також, без комп'ютерної допомоги, процес формування заявки і прийняття рішення по резервному включенню займає тривалий час. Беручи до уваги всі вище викладені фактори, можна переконатися в тому, що без комп'ютерних обчислювальних систем важко домогтися налагодженого і своєчасного реагування на аварії.

Шлях до вирішення поставленої задачі лежить через:

– опис інтелектуальної діяльності оперативно-диспетчерського персоналу з аналізу та перепідключення електричних мереж, її структурування для подальшого моделювання;

– наповнення контентом і моделювання кожного процесу окремо, по відношенню до даної предметної області;

– формалізація логіки послідовності дій, при перемиканні в ланцюгах;

– виключення можливості підключення в підстанцію двох фідерів;

– впровадження ЕС (експертної системи) на персональний комп'ютер (ПК) з розробкою графічного інтерфейсу.

У роботі застосовуються методи теорії графів, множин, формальної математичної логіки.

Наукова новизна полягає у наступному:

– предметна область формалізується (аналіз і повторна комутація в електричній мережі) з вибором процесів, які моделюють інтелектуальну діяльність диспетчера, при прийнятті рішень, структури даних і знань;

– моделі базових інтелектуальних процесів для ЕС з їх реалізацією;

– розроблено оригінальні методи та обчислювальні процедури для структурного аналізу стану комутації електричних мереж:

а) відсутність підключення двох фідерів на станцію;

б) перспектива розширення моделі;

в) уміння працювати з великими обсягами інформації;

г) можливість ручного управління;

д) облік часу роботи підстанцій.

Практична цінність основана на таких тезах:

– застосування запропонованого підходу до формалізації логічного виводу на базі експертної системи, що може дозволити розробити систему комутації, що володіє інваріантними властивостями по відношенню до мереж різних рівнів складності;

– експертна система перемикання наділена одною з основних властивостей, що повинні бути притаманні будь-якій експертній системі – декларативність («відкритість», здатність до навчання, доступність);

– використання таких систем в промисловості звільнить диспетчера від рутинної роботи і зменшить кількість допущених помилок;

– вони можуть бути використані в навчальних системах-симуляторах оперативного перемикання;

– при проектуванні наступних підстанцій можуть вказувати на найбільш вигідне географічне розташування.

1 АНАЛІЗ СУЧАСНОГО СТАНУ ЕКСПЕРТНИХ СИСТЕМ В ОБЛАСТІ ДІАГНОСТИКИ

1.1 Поняття інтелектуальної інформаційної системи

Інтелект – пізнання, розуміння, осягнення, сукупність розумових здібностей людини. Відповідно, штучний інтелект визначається, як здатність автоматизованих систем виконувати деякі функції інтелекту людини, наприклад, знаходити і приймати оптимальні рішення, ґрунтуючись на отриманому раніше досвіді і раціональному аналізі зовнішніх факторів. Також під інтелектом розуміється здатність мозку вирішувати складні завдання, набуваючи, запам'ятовуючи і цілеспрямовано перетворюючи знання, в процесі навчання на досвід, в результаті адаптації до різних ситуацій.

Залежно від своєї природи, знання буває фактуальне та операційне. Фактуальне знання – осмислені дані. Операційне знання – загальні залежності між фактами, які дозволяють інтерпретувати дані або витягувати з них нову інформацію.

До головних недоліків традиційної інформаційної системи (ІС) відносяться:

– слабка адаптованість до інформаційних потреб користувача;

– неможливість вирішувати погано формалізовані завдання.

Виділені недоліки усуваються в інтелектуальних інформаційних системах.

ІІС – це інформаційна система, яка заснована на концепції використання бази знань для генерації алгоритмів розв'язання задач різних класів, в залежності від конкретних інформаційних потреб користувача.

ІІС мають такі характерні ознаки:

– розвинені комунікативні здібності;

– вміння вирішувати складні, погано формалізовані

завдання (характеризуються наполовину якісним і кількісним описом, а добре формалізовані завдання – повністю кількісним описом);

– здатність до розвитку і самонавчання.

Умовно кожному з цих ознак відповідає свій клас ІІС. Класифікацію ІІС зображено на рисунку 1.1.

Системи з інтелектуальним інтерфейсом	Експертні системи	Самонавчальні системи	Адаптивні інформаційні системи
- інтелектуальні бази даних	- класифікуючі	- індуктивні системи	- CASE технології
- природно-мовний інтерфейс	- довизначаюча	- нейронні мережі	- компонентні технології
- гіпертекстові системи	- трансформуючі	- системи, засновані на прецедентах	
- системи контекстної допомоги			
- системи когнітивної графіки	- мультиагентні	- інформаційні сховища	

Рисунок 1.1 – Класифікація інтелектуальних інформаційних систем

1.2 Відмінності експертних систем від інших видів програм штучного інтелекту

1.2.1 Системи з інтелектуальним інтерфейсом

Перший клас систем – системи з інтелектуальним інтерфейсом, що базуються на комунікативних здібностях.

Інтелектуальні бази даних (БД) дозволяють, на відміну від традиційних, забезпечувати вибірку необхідної інформації, що не присутня в явному вигляді, а є висновком сукупності даних, що зберігаються.

Природно-мовний інтерфейс (ПМ-інтерфейс) застосовується для доступу до інтелектуальної БД, контекстного пошуку документальної текстової інформації, голосового введення команд в системах управління, машинного перекладу з іноземних мов. На відміну від аналізу, синтез висловлювань полягає в перетворенні цифрового подання інформації в уявлення природньою мовою.

Гіпертекстові системи використовуються для реалізації пошуку за ключовими словами в БД з текстовою інформацією.

Системи контекстної допомоги – відносяться до класу систем поширення знань. Такі системи є, як правило, додатками до документації. Системи контекстної допомоги – окремий випадок гіпертекстових ПМ-систем. У них користувач описує проблему, а система на основі додаткового діалогу конкретизує її і виконує пошук ситуації, що стосується рекомендацій.

Системи когнітивної графіки орієнтовані на спілкування з користувачем ІІС засобами графічних образів, які генеруються відповідно до змін параметрів модельованих процесів, що спостерігаються. Когнітивна графіка дозволяє в наочному і виразному вигляді уявити безліч параметрів, що характеризують досліджуване явище.

1.2.2 Експертні системи

ЕС займаються рішенням складних завдань.

Класифікуюча ЕС вирішує завдання розпізнавання ситуацій. Основним методом формування рішень в таких системах є дедуктивний логічний висновок.

Довизначаюча ЕС використовується для вирішення завдань з неповністю визначеними даними і знаннями. У таких ЕС виникають завдання інтерпретації нечітких знань і вибору альтернативних напрямків пошуку в просторі можливих рішень. В якості методів обробки

невизначених знань можуть використовуватися: Баєсовський імовірнісний підхід, коефіцієнти впевненості, нечітка логіка.

Трансформуюча ЕС відноситься до синтезуючим динамічним експертним системам, в яких передбачається повторююче перетворення знань, в процесі вирішення завдань. В ЕС даного класу використовуються різні способи обробки знань:

– генерація і перевірка гіпотез;

– логіка припущень і замовчувань (коли за неповними даними формуються уявлення про об'єкти певного класу, які згодом адаптуються до конкретних умов, що змінюються);

– використання метазнань (більш загальних закономірностей) для усунення невизначеностей.

Мультиагентні системи – це динамічні ЕС, засновані на інтеграції декількох різнорідних джерел знань. Ці джерела обмінюються між собою одержуваними результатами в ході вирішення завдань. Системи даного класу мають такі можливості:

– реалізація альтернативних міркувань на основі використання різних джерел знань і механізму усунення протиріч;

– розподілене рішення проблем, що декомпозуються на паралельно вирішувані підзадачі з самостійними джерелами знань;

– застосування різних стратегій виведення висновків залежно від типу проблем, що вирішуються;

– обробка великих масивів інформації з баз даних;

– використання математичних моделей і зовнішніх процедур для імітації розв’язку ситуацій.

1.2.3 Самонавчальні системи

Характеризуються здатністю до самонавчання. Індуктивні системи дозволяють узагальнювати приклади на основі принципу індукції «від

часткового до загального». Процедура узагальнення зводиться до класифікації прикладів по значимим ознаками.

Нейронні мережі є класичним представником технології, заснованої на прикладах. Нейронні мережі – узагальнена назва групи математичних алгоритмів, що володіють здатністю навчатися на прикладах, «впізнаючи» згодом риси зразків і ситуацій, що попередньо були побачені. Завдяки цій здатності нейронні мережі використовуються при вирішенні задач обробки сигналів і зображень, розпізнавання образів, а також для прогнозування.

У системах, заснованих на прецедентах, база знань (БЗ) містить опис конкретних ситуацій (прецедентів). Прецеденти описуються безліччю ознак, за якими будуються індекси швидкого пошуку. Однак в системах, заснованих на прецедентах, на відміну від індуктивних систем, допускається нечіткий пошук з отриманням безлічі допустимих альтернатив, кожна з яких оцінюється деяким коефіцієнтом впевненості. Системи, засновані на прецедентах, застосовуються для поширення знань і в системах контекстної допомоги.

Інформаційні сховища відрізняються від інтелектуальних баз даних, тим, що представляють собою сховища значимої інформації, що регулярно витягується з оперативних баз даних. Сховище даних – це предметно-орієнтований, інтегрований, прив'язаний до часу незмінний набір даних, що використовується для підтримки процесів прийняття управлінських рішень.

1.2.4 Адаптивні інформаційні системи

Потреба в адаптивних інформаційних системах виникає в тих випадках, коли підтримувані ними проблемні області постійно розвиваються. У зв'язку з цим, адаптивні системи повинні задовольняти ряд специфічних вимог, а саме:

– адекватно відображати знання проблемної області в кожен момент часу;

– бути придатними для легкої і швидкої реконструкції при зміні проблемного середовища.

Адаптивні властивості інформаційних систем забезпечуються за рахунок інтелектуалізації їх архітектури. Ядром таких систем є модель проблемної області, що постійно розвивається, підтримується в спеціальній базі знань – репозиторії. Ядро системи керує процесами генерації та конфігурації програмного забезпечення.

1.3 Особливості експертних систем

Інтелектуальна інформаційна система володіє знаннями і використовується для досягнення поставленої мети. ІС не може бути без знань. Експертні системи є дійсно інтелектуальними системами і, врешті-решт, інтелектуальність визначила їх комерційний успіх. Інтелектуальна система буде тоді інтелектуальною, коли її забезпечать спеціальними знаннями з необхідної предметної області.

Експертна система повинна бути корисна не тільки професіоналам, але і звичайній людині, чия сфера діяльності, ні як не пов'язана з діагностикою, профілактичними роботами, аналізом повідомлень від засобів обчислювальної техніки. Для таких людей ЕС є помічником в їх повсякденній діяльності. Взаємодія людини з ЕС повинна бути настільки простою, як в повсякденному житті люди користуються телевізором, радіо, машиною і самим комп'ютером.

В реальній життєвій ситуації експертні системи застосовують в якості радника, де професіонал сумнівається у виборі, або не має достатнього часу для вибору правильного шляху вирішення проблеми.

Для того, щоб експертна система працювала ефективно, вона володіє характеристиками:

– пошуку правильного рішення складних завдань за певний проміжок часу;

– вирішення вузькоспеціалізованих проблем, застосовуючи особливі прийоми, які дозволяють уникати сліпого пошуку;

– будування міркування, маніпулюючи символьним описом, тобто передбачаючи, що будь-яку фізичну систему можна схематично описати (концептуалізувати) за допомогою чітких понять, і представити цей опис за допомогою символів;

– здатна приводити первісну похідну постановку задачі до виду, придатному для обробки за правилами, які були прийняті в даній експертній системі. За допомогою такого формулювання завдання, експертна система знає межі своїх можливостей і не буде, в щоб те не стало, втискати цю задачу в зручні для себе рамки моделей;

– експертна система завжди прозора, в тому сенсі, що вона не тільки відповідає на задані їй питання, а й пояснює хід міркування, що веде до відповіді;

– наведені характеристики в експертній системі забезпечують інтерактивний інтерфейс з користувачем, що призначений для обґрунтування своїх висновків.

Виходячи з вище зазначеного можна дійти висновку, що експертна система містить знання з предметної області, які були накопичені в результаті практичної діяльності людей, і використовується для вирішення проблем, з певній галузі діяльності.

1.4 Переваги використання експертних систем

При використанні ЕС виникає питання: «Навіщо розробляти експертні системи. І чи не краще звернутися до людського досвіду, як це було в минулому?». Відзначимо лише основні переваги, які дає використання ЕС.

Перевагами використання ЕС є:

– сталість. Людська компетенція, слабшає з часом. Перерва в діяльності людини-експерта може серйозно відбитися на його професійній якості;

– легкість передачі або відтворення. Передача знань від однієї людини іншій – довгий і дорогий процес. Передача штучної інформації – це простий процес копіювання програми або файлу даних;

– стійкість і відтворюваність результатів. Експерт-людина може приймати в різних ситуаціях різні рішення через емоційні чинники. Результати ЕС – стабільні;

– вартість. Експерти, особливо висококваліфіковані коштують дуже дорого. ЕС, навпаки, порівняно недорогі. Їх розробка дорога, але вони дешеві в експлуатації та обслуговуванні.

Розробка ЕС не дозволить відмовитися від фахівця електромережевого сектору. Незважаючи на те, що експертна система добре справляється зі своєю роботою, тим не менше, бувають такі ситуації, де експертна система не зможе замінити людину.

Однак і в цих випадках ЕС може дозволити відмовитися від послуг висококваліфікованого експерта. Або, точніше кажучи, перенести сферу його діяльності з виконання автоматичних завдань до інформаційного аналізу. Так як користувач самостійно зможе провести діагностику аварійної ситуації без спеціальної підготовки, використовуючи експертну систему.

1.5 Області використання ЕС

Процес створення експертної системи, на перших етапах, полягає у специфічній взаємодії експерта (експертів) і інженера по знаннях, з метою «вилучення» з експерта і вбудовування в систему процедур, стратегій, емпіричних правил, які він використовує для вирішення завдань.

Експерт – це людина, яка, завдяки навчанню і досвіду, може робити те, що решта людей робити не вміють. Експерти працюють не просто професійно, але швидко і ефективно. Вони добре вміють розпізнавати проблеми, з якими стикаються, за прикладом тих типових проблем, з якими вони вже знайомі. Дуже важливо підкреслити, що експерт повинен не тільки знати, а й уміти.

Основним предметом дослідження є знання, їх придбання, представлення та використання. Фахівці, які займаються цим, використовують термін «інженерія знань». Ця область науки пов'язана з розробкою експертних систем, баз знань і баз даних; вивчає етапи аналізу і розробки експертних систем, що зображені на рисунку 1.2; вивчає методи і засоби вилучення, уявлення, структурування і використання знань.

Про експертну систему можна сказати наступне, експертна система – це обчислювальна система, в яку включені знання фахівців про деяку вузьку предметну область в формі бази знань. Експертні системи приймають або допомагають прийняти рішення замість фахівця.

Характерними рисами експертної системи є:

– чітка обмеженість предметної області;

– здатність приймати рішення в умовах невизначеності;

– здатність пояснювати хід і результат рішення зрозумілим для користувача способом;

– чіткий поділ декларативних і процедурних знань (фактів і механізмів виведення);

– здатність поповнювати базу знань, можливість нарощування системи;

– вивід результату у вигляді конкретних рекомендацій, певного набору дій в ситуації, що склалася, не поступаючись рішенням кращих фахівців;

– орієнтація на рішення неформалізованих (спосіб формалізації поки невідомий) завдань.

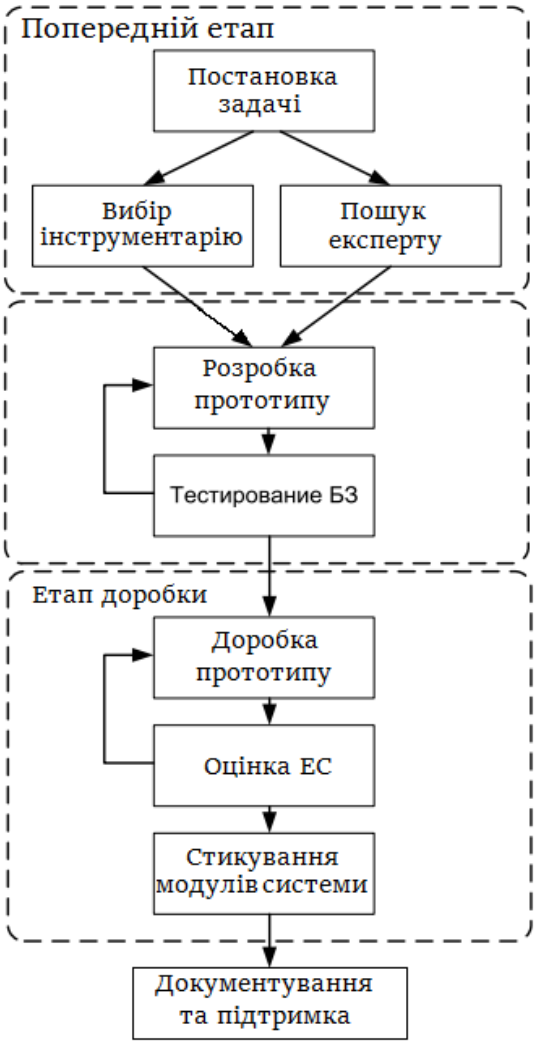

Рисунок 1.2 – Етапи розробки експертної системи

ЕС відноситься до таких видів систем штучного інтелекту, які широко поширені і знайшли практичне застосування. Експертні системи використовують при вирішенні серйозних завдань. За якістю і ефективністю вирішення, експертні системи не поступаються роботі людині, що професійно займається, наприклад, діагностикою та пошуком несправностей.

При вирішенні задач, експертні системи мають бути прозорими, тобто мати функціонал, метою якого є пояснення користувачеві хід та результат взаємодії з інтерфейсом на високому рівні (на відміну від рішень, отриманих за допомогою числових алгоритмів, і особливо від рішень

отриманих статистичними методами). Експертні системи здатні поповнювати свої знання в ході взаємодії з експертом.

В даний час експертні системи застосовують при вирішенні різних типів завдань: інтерпретація, прогноз, діагностика, планування, конструювання, контроль, налагодження, інструктаж, управління. У самих різних областях людського життя, такі як: фінанси, нафтова і газова промисловість, енергетика, транспорт, фармацевтичне виробництво, космос, металургія, гірнича справа, хімія, освіта, целюлозно-паперова промисловість, телекомунікації і зв'язок і ін.

Експертні системи мають справу з предметами реального світу, операції з якими зазвичай вимагають наявності значного досвіду, накопиченого людиною. Безліч програм з області штучного інтелекту є суто дослідницькими, і основна увага в них приділяється абстрактним математичним проблемам, або спрощеним варіантам реальних проблем (іноді їх називають «іграшковими» проблемами), а метою виконання такої програми є «підвищення рівня інтуїції», або відпрацювання методики. Експертні системи мають яскраво виражену практичну спрямованість в науковій або комерційній області.

Однією з основних характеристик експертної системи є її продуктивність, тобто швидкість отримання результату і його достовірність (надійність).

Дослідницькі програми штучного інтелекту можуть і не бути дуже швидкими, можна примиритися і з існуванням в них відмов в окремих ситуаціях, оскільки, врешті-решт, – це інструмент дослідження, а не програмний продукт. А ось експертна система повинна за прийнятний час знайти рішення, яке було б не гірше, ніж те, яке може запропонувати фахівець в цій предметній області.

Експертна система повинна мати здатність пояснити, чому запропоновано саме таке рішення, і довести його обґрунтованість. Користувач повинен отримати всю інформацію, необхідну йому для того,

щоб бути впевненим, що рішення прийнято «не зі стелі». На відміну від цього, дослідницькі програми «спілкуються» тільки зі своїм творцем, який і так (швидше за все) знає, на чому ґрунтується її результат. Експертна система проектується з розрахунку на взаємодію з різними користувачами, для яких її робота повинна бути, по можливості, прозорою.

Найчастіше термін система, заснована на знаннях (knowledge-based system), використовується, як синонім терміну експертна система, тобто експертна система це більш широке поняття. Система, заснована на знаннях – це будь-яка система, процес роботи якої заснований на застосуванні правил відносин до символічного подання знань, а не на використанні алгоритмічних або статистичних методів. Таким чином, програма, що здатна міркувати про погоду, буде системою, заснованою на знаннях, навіть в тому випадку, якщо вона не здатна виконати метеорологічну експертизу. А ось щоб мати право називатися метеорологічної експертної системою, програма повинна бути здатна давати прогноз погоди (інше питання наскільки він буде достовірний).

Підсумовуючи вище сказане, зазначимо – експертна система містить знання в певній предметній області, накопичені в результаті практичної діяльності людини (або людства), і використовує їх для вирішення проблем, специфічних для цієї області. Цим експертні системи відрізняються від інших, «традиційних» систем, в яких перевага віддається більш загальним, але менш пов'язаним з предметною областю теоретичним методам, найчастіше математичним. Процес створення експертної системи часто називають інженерією знань (knowledge engineering), і він розглядається в якості «застосування методів штучного інтелекту».

1.6 Функції експертної системи

Не всяку систему, засновану на знаннях, можна розглядати як експертну.

Експертна система повинна також уміти якимось чином пояснювати свою поведінку і свої рішення користувачу, так само, як це робить експерт-людина. Це особливо необхідно в областях, для яких характерна невизначеність, неточність інформації (наприклад, у медичній діагностиці). У цих випадках, здатність до пояснення потрібна для того, щоб підвищити ступінь довіри користувача до інструментарію системи, а також для того, щоб дати можливість користувачу знайти можливий дефект у міркуваннях системи. У зв'язку з цим, в експертних системах варто передбачати дружню взаємодію з користувачем, що робить для користувача процес міркування системи «прозорим».

Часто до експертних систем висувають додаткову вимогу – здатність мати справу з невизначеністю і неповнотою. Інформація про поставлену задачу може бути неповною або ненадійною; відносини між об'єктами предметної області можуть бути наближеними. Наприклад, може не бути повної впевненості в наявності у пацієнта деякого симптому, чи в тому, що дані, отримані при вимірюванні деякої фізичної чи біологічної характеристики – вірні. Ліки можуть стати причиною ускладнення, хоча зазвичай цього не відбувається. У всіх цих випадках необхідні міркування з використанням імовірнісного підходу.

У найзагальнішому випадку для того, щоб побудувати експертну систему, ми повинні розробити механізми виконання наступних функцій системи:

– рішення задач з використанням знань про конкретну предметну область. Можливо, при цьому виникне необхідність мати справу з невизначеністю;

– взаємодія з користувачем, включаючи пояснення намірів і рішень системи, під час і після закінчення процесу рішення задачі.

Кожна з цих функцій може виявитися дуже складною, і залежить від прикладної області, а також від різних практичних вимог. В процесі розробки і реалізації можуть виникати різноманітні важкі проблеми. Тут ми

обмежимося позначками основних ідей, які підлягають, в подальшому, деталізації та вдосконалення.

1.7 Структура та режими використання ЕС

Типова ЕС складається з наступних основних компонентів: вирішувач (інтерпретатор), робочої пам'яті (РП), яку називають також базою даних (БД), бази знань (БЗ), компонентів набування знань, пояснювального і діалогового компонентів, що показано на рисунку 1.3.

База даних призначена для зберігання вихідних і проміжних даних, розв'язуваного в поточний момент завдання. Цей термін збігається за назвою, але не за змістом з терміном, що застосовується в інформаційно-пошукових системах (ІПС) і системах управління базами даних (СУБД) для позначення всіх даних (і в першу чергу – не поточних, а довгострокових), що зберігаються в системі.

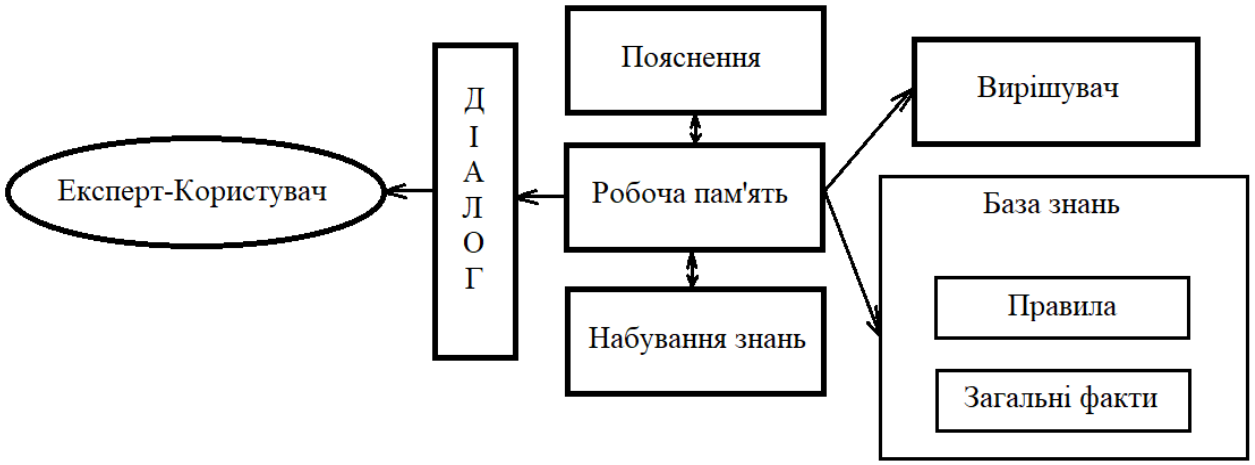

Рисунок 1.3 – Типова структура експертної системи

База знань в ЕС призначена для зберігання довгострокових даних, що описують розглянуту область, а не поточних даних і правил, що описують доцільні перетворення даних цієї області.

Вирішувач, використовуючи вихідні дані з РП та знань з БЗ, формує

таку послідовність правил, які, будучи застосованими до вихідних даних, приводять до рішення задачі.

Пояснювальний компонент пояснює, як система отримала розв'язок задачі (або чому вона не отримала рішення), і які знання вона при цьому використовувала. Це полегшує експерту тестування системи і підвищує довіру користувача до отриманого результату.

Діалоговий компонент має бути орієнтований на організацію дружнього спілкування з усіма категоріями користувачів, як в ході вирішення завдань, так і в ході придбання знань, пояснення результатів роботи.

Знання в експертній системі можуть бути представлені безліччю способів. Одним з широко застосовуваних методів представлення знань є правила в формі “IF THEN”. Його сутність полягає в тому, що в разі виявлення факту, відповідним шаблоном правила, має бути виконана, обумовлена правилом дія.

В даному прикладі шаблоном є фраза «людина не має постійної роботи». У разі, коли факт відповідає шаблоном – рекомендується виконати дію «кредит не видавати».

IF (Людина не має постійної роботи)

THEN { Кредит не видавати }

Застосовуючи підхід до подання знань експертів у вигляді правил, було розроблено кілька успішних експертних систем. До них можна віднести систему XCON/R1, створену в компанії Digital Equipment Corporation (DEC), яка здатна проектувати конфігурації комп'ютерних чи електричних систем, не поступаючись окремим експертам-людям.

Даний підхід виявився також прийнятним для створення значної кількості порівняно невеликих систем, з метою вирішення спеціалізованих завдань на основі декількох сотень правил. Такі системи не можуть функціонувати на рівні експерта, проте дозволяють застосовувати інтелектуальні технології для вирішення завдань, які потребують великих

обсягів знань. Знання, необхідні для подібних невеликих систем, можуть бути отримані з книг, журналів або з іншого джерела загальнодоступної інформації.

1.8 Класифікація ЕС

Для класифікації ЕС використовують такі ознаки:

– спосіб формування рішення;
– спосіб обліку тимчасової ознаки;
– вид використовуваних даних;
– число використовуваних джерел рішення знань.

За способом формування рішення, ЕС можна розділити на аналізуючі та синтезуючі. У системах першого типу здійснюється вибір рішення з багатьох відомих рішень на основі аналізу знань, в системах другого типу рішення –синтезується з окремих фрагментів знань.

Залежно від способу обліку тимчасової ознаки, ЕС ділять на статичні і динамічні. Статичні ЕС призначені для вирішення завдань з незмінними, в процесі вирішення, даними і знаннями, а динамічні ЕС – допускають такі зміни.

За видами використовуваних даних і знань розрізняють ЕС з детермінованими і невизначеними знаннями. Під невизначеністю знань і даних розуміється їх неповнота, ненадійність, нечіткість.

ЕС можуть створюватися з використанням одного або декількох джерел знань.

1.9 Діагностика знань та умінь за допомогою експертних систем

У широкому сенсі, діагностикою називають процес пошуку несправності в системі, заснованій на інтерпретації даних. Стосовно до процесу навчання, діагностика означає виявлення тих знань / умінь, якими

учень не опанував.

При побудові ЕС, призначених для діагностики поточного стану знань / умінь учня, можливі два підходи. Перший підхід підпорядкований логіці розкриття теми. При цьому підбираються завдання (питання), сумарний спектр яких, тобто сукупність всіх знань / умінь, необхідних для вирішення вихідної задачі, покриває відповідну частину предметної моделі учня (по області, що підлягає діагностиці).

Другий підхід полягає в тому, що предметна модель учня обмежується спектром будь-якої задачі або набору задач. Другий підхід підпорядкований логіці вирішення певної задачі. Якщо учень правильно вирішує завдання, то вважається, що поточна модель учня збігається з предметною і не потрібує ніяких коректувань навчального процесу. Якщо допущена помилка, то завдання діагностуючої системи полягає у визначенні тих знань / умінь, якими той, кого навчають, не опанував.

Принциповим, з нашої точки зору, є питання про вибір виду гіпотез. Як зазначалося вище, стосовно навчання, діагностика передбачає виявлення розділів або окремих понять / умінь навчальної дисципліни, володіння якими з боку учня незадовільно.

При побудові діагностуючої ЕС пропонується методика, яка полягає в наступному. Сформулювати гіпотези виду «той, якого навчають не знає / не вміє», а для оцінки впливу симптому S на гіпотези цього виду означити цей симптом, використовуючи шкалу «Так - Не знаю – Ні». Якщо той, якого навчають виконує дію неправильно (при цьому значення симптому S «Ні»), то апостеріорна імовірність гіпотези зростає. Ця гіпотеза може бути прийнята, і система виведе повідомлення, що дозволяє діагностувати знання / уміння учня. Якщо той, кого навчають відмовився відповідати на питання, запропоноване ЕС (при цьому значення симптому S – «Не знаю»), то методика передбачає, що апостеріорна ймовірність гіпотези дорівнює її апріорній імовірності, і траєкторія обчислень призводить до невизначеності гіпотези.

Описаний підхід має істотні недоліки. По-перше, відповідь «Не знаю» має два значення. Вона може бути дана: коли учень дійсно не знає відповідь, або у разі, якщо в студента не вистачає часу для обдумування. По-друге, в разі правильного рішення задачі учнем (при цьому значення симптому S – «Так»), гіпотеза виду «той, якого навчають не знає / не вміє» відкидається, а іншої гіпотези немає.

Зазначені вище недоліки методики призводять до того, що система, побудована відповідно до даної методики, як при правильному вирішенні учнем завдання, так і в разі відмови учня від відповіді – видає одне і те саме повідомлення: «Немає даних для прийняття рішення чи немає проблем!» Але це повідомлення двозначно. Перша частина повідомлення говорить про те, що гіпотези залишилися актуальними, але їх апостеріорні імовірності не перевищили верхній поріг, і вони не можуть бути прийняті. Друга ж частина стверджує, що всі гіпотези відкинуті. Як показано вище, дії учня, що викликають це повідомлення, можуть бути принципово різними. Повідомлення, що видається ЕС, не дозволяє викладачеві, не звертаючись до підсистеми пояснення прийняття рішення або протоколу експертизи, визначити, які дії учня привели до даного результату. Звернення до підсистеми пояснення дозволяє відновити хід міркувань учня, однак цей спосіб є дуже громіздким і вимагає великих витрат часу.

Для того, щоб виключити таку ситуацію, методика побудови ЕС була модернізована. По-перше, виключається випадок нестачі часу для обдумування. Це досягається тим, що час, необхідний для роботи з системою, визначається експериментально. Це робиться в такий спосіб: складаються картки, що включають ті ж самі питання і завдання, якими буде наповнена база знань ЕС.

За допомогою цих карток проводиться тестування. Оператору системи для відповідей дається необмежений час. Визначається час, за який всі учні впораються із завданням. Цей час вважається в подальшому також часом для роботи з ЕС. Тому відповідь «Не знаю» розглядається тільки як

нездатність учня відповісти на питання, що запропоновано ЕС.

Правильне рішення задачі призводить до зменшення імовірності гіпотез виду «той, якого навчають не знає / не вміє», що залежать від даного симптому, і збільшення ймовірності гіпотез виду «той, якого навчають знає / вміє», і навпаки. При цьому ми вважаємо, що система, побудована таким чином, є діагностуючою, оскільки вона виявляє знання / незнання (вміння / невміння) учнем окремих питань, в той час як тестування, наприклад, передбачає інтегральну оцінку рівня знань учня.

ЕС можуть виступати як інструмент навчання методам і способам виділення головного на навчальному матеріалі. Вони можуть використовуватися як для традиційної діагностики знань / умінь учня з його відповідей на запропоновані системою питання, так і для діагностики знань / умінь учня з тих питань, які він становить при роботі з реальною базою знань ЕС.

1.10 Обмеження в застосування експертних систем

Навіть найкращі з існуючих ЕС, які ефективно функціонують як на великих, так і на міні-ЕОМ, мають певні обмеження в порівнянні з людиною-експертом, такі як:

– більшість ЕС не цілком придатні для застосування кінцевим користувачем. Якщо ви не маєте деякого досвіду роботи з такими системами, то у вас можуть виникнути серйозні труднощі. Багато системи виявляються доступними лише тим експертам, які створювали базу знань;

– режим «Питання-Відповідь», зазвичай прийнятий в таких системах, уповільнює отримання рішень. Наприклад, без системи MYCIN, лікар може (а часто і повинен) прийняти рішення значно швидше, ніж з її допомогою;

– навички системи не зростають після сеансу експертизи;

– все ще залишається проблемою приведення знань, отриманих від

експерта, до виду, що забезпечує їх ефективну машинну реалізацію;

– ЕС не здатні навчатися, не володіють здоровим глуздом. Домашні кішки здатні навчатися навіть без спеціального дресирування, дитина в стані легко усвідомити, що він стане мокрим, якщо перекине на себе склянку з водою, проте якщо почати виливати каву на клавіатуру комп'ютера, у нього не вистачить «розуму» відсунути її;

– ЕС незастосовні у великих предметних областях. Їх використання обмежується предметними областями, в яких експерт може прийняти рішення за час від декількох хвилин до декількох годин;

– у тих областях, де відсутні експерти (наприклад, в астрології), застосування ЕС виявляється неможливим;

– має сенс залучати ЕС тільки для вирішення когнітивних завдань. Теніс, їзда на велосипеді не можуть бути предметною областю для ЕС, однак такі системи можна використовувати при формуванні футбольних команд;

– людина-експерт при вирішенні задач зазвичай звертається до своєї інтуїції або здорового глузду, якщо відсутні формальні методи рішення або аналоги таких завдань.

Системи, засновані на знаннях, виявляються неефективними при необхідності проведення скрупульозного аналізу, коли число «рішень» залежить від тисяч різних факторів і багатьох змінних, які змінюються в часі. У таких випадках краще використовувати бази даних з інтерфейсом природною мовою.

2 ВИКОРИСТАННЯ ЕКСПЕРТНОЇ СИСТЕМИ В ЕЛЕКТРОЕНЕРГЕТИЦІ

2.1 Роль експертних систем в електроенергетиці

Метою створення експертної системи є автоматизація процесу прийняття рішень. Останнім часом з'явився такий напрямок в розробці методів прийняття рішень, як наближення опису та формалізації проблеми до природньої людської мови і розуміння. Диспетчеру системи часто буває складно конкретно оцінити об'єкт по якомусь критерію. Диспетчер може сумніватися і буде шукати середній рейтинг. Часто такі труднощі виникають не через нестачу досвіду, а як раз навпаки. Згодом виникла потреба в розробці методів, гнучких для сприйняття людиною інформації, що враховують невизначеність у великій кількості вимірів.

В основі ЕС лежить база знань (БЗ) про конкретну предметну область, яка повинна накопичуватися в результаті побудови та експлуатації ЕС. Накопичена інформація та організація її подання для людини – є найважливішою перевагою використання ЕС.

Інформація не завжди може бути подана точно, існує так звана «досить чітка інформація» або «нечіткі» знання. Правила обробки та подання такої інформації базується на логіці. Коли виникає ситуація, така, що для прийняття рішення або подання інформації не вистачає обсягу, або повністю відсутня необхідна інформація, раціональне рішення може бути прийнято завдяки накопиченим знанням, в цьому і полягає головні переваги міркування імовірнісного над логічним. Це можливо завдяки накопиченню знань в базі даних, в результаті роботи експертної системи, після чого, експертна система, базуючись на отриманих раніше знаннях, буде приймати рішення ґрунтуючись на найбільш підходящій ймовірності, так званому - суб'єктивному довірі. Так як, прийняті рішення носять імовірнісний характер, експертні системи ґрунтуються на теорії ймовірності. Саме тому, більшість експертних систем є «Баєсовими».

Для процесу управління можливостями невизначеності, теорія Баєса є – основною в теорії ймовірності, формула 2.1.

$$P(A/B) = \frac{\mathrm{P}(B/A)*\mathrm{P}(\mathrm{A})}{\mathrm{P}(B/A)*P(A)+\mathrm{P}(B/\bar{A})*P(\bar{A})} \quad (2.1)$$

де А – є деякою подією;

В – друга подія;

Р (В/А) – умовна імовірність, імовірність виникнення події В за умови, що подія А вже відбулася.

Дане правило надає спосіб для отримання умовної імовірності події В за умови А.

Дане відношення дозволяє експертній системі керувати невизначеністю. Якщо використовувати цю формулу, то можна зробити висновок про те, що: для того, щоб зменшити кількість умовних імовірностей, які повинні бути визначені, необхідно, встановити зв'язки між умовно незалежними змінними, які можуть полегшувати обчислення результатів запитів.

2.2 Проведення процедури прийняття рішення

Яким чином експертна система приймає рішення? Для того, щоб ЕС змогла об'єктивно оцінити ситуацію і прийняти рішення, слід чітко прописати в алгоритмі поставлені цілі і завдання, також необхідно сформулювати критерії оптимального управління. Сама ЕС оперуючи набором даних, буде вибирати найбільш підходящий варіант з цілого переліку критеріїв, що знаходяться в базі даних.

Всі дані, які необхідні для коректної роботи ЕС можна розділити на 4 групи:

– інформація про мету і завдання;

– інформація про критерії оптимального управління;
– інформація про можливі альтернативні варіанти;
– інформація про найбільш зручний вибір.

Розберемо деякі з цих груп. Після створення ЕС під конкретні цілі та завдання, алгоритм розуміє, яку інформацію йому необхідно вибирати для прийняття рішення.

Наступним кроком буде відбір інформації. ЕС порівнює всі дані з критеріями, які прописані в алгоритмі. На даному етапі, завданням ЕС ставиться вибір найбільш підходящого критерію. Варто відзначити, що існують завдання багатокритеріального відбору, в таких завданнях ЕС доводиться вибирати не тільки одне рішення, а цілий пакет рішень. У таких ситуаціях існує деяка складність у вигляді неможливості вибору найкращого рішення. У цьому випадку, ЕС спирається на альтернативні варіанти і накопичену інформацію в результаті своєї роботи.

2.3 Формалізація логіки оперативних перемикань в електричних мережах

У сучасному світі, рівень автоматизації на диспетчерських пунктах залишається вкрай низьким. Наприклад, на сьогоднішній день, диспетчера працюють за наступним алгоритмом. Диспетчеру служби диспетчерського та технологічного управління по телефону, або електронній пошті надходить запит на ремонт повітряних ліній електропередач. Далі, диспетчер заповнює бланк для подачі заявок, вносить необхідні дані, наприклад, дані про об'єкт, час виконання робіт, дату надходження і так далі. Потім він відправляє заявки на розгляд запиту, виконання або відмову у виконанні заявки. Після зміни статусу виконання заявки, співробітник, згідно з правилами ведення журналу заявок, вносить інформацію про неї в журнал заявок, куди записує номер заявки, ПІБ диспетчера, дату надходження, появу відмови, якщо існує, а також статус заявки. Потім

диспетчер відправляє заявку в архів. І це не враховуючи того, що диспетчер часто сам приймає рішення яка підстанція буде резервувати поточну поломку.

Очевидно, що обраний альтернативний шлях, повинен бути економічно вигідним, більш того, диспетчеру необхідно врахувати і супутні фактори:

– режим роботи підстанції;

– необхідна потужність;

– пріоритет підстанції.

Наприклад, паротурбінна ТЕЦ буде обрана, якщо з міркувань надійності та технічним характеристикам необхідно забезпечувати потужність не менше декількох мегават при тривалій роботі.

У разі, якщо навантаження на підстанції може змінюватися швидко, то найімовірніше буде обрана дизельна електроцентраль або двигуни швидкого запуску.

У промислових масштабах, постає питання про забезпечення безперебійного живлення електрообладнання, так як існує особлива, що класифікується як перша, група електрообладнання за необхідною надійністю живлення. Дана група обладнання не може стояти без напруги тривалий час, так як це може призвести або до надзвичайних ситуацій, або дуже дорогому повторному запуску технологічного процесу. Для такого обладнання, застосовують високонадійні автономні місцеві джерела.

Місцевими джерелами напруги прийнято вважати:

– синхронні генератори заводських ТЕЦ та інших регулярно працюючих заводських електростанцій і електричних генераторних установок;

– синхронні двигуни з cosφ 0,9;

– конденсаторні батареї.

В результаті, можна зробити висновок, що при розробці ЕС необхідно формалізувати логіку нечітких даних (знань), які повинні стати основою

роботи алгоритму і лягти в основу процедур, виконуваних експертною системою. Під знаннями розуміються дані і правила, за якими відбувається комутація і перемикання пристроїв.

На сьогоднішній день, при розробці експертної системи, для перемикання розподільної електричної мережі, використовують так званий «закритий» спосіб, який створює певні складності. У разі зміни умов експлуатації або зміни характеристик об'єкта управління, неможливо змінити логіку роботи експертної системи без переробки всієї логіки. Даний недолік багато в чому пов'язаний як правило з тим, що експертна система заточена саме під конкретну задачу, в яку входить:

– аналіз аварії;

– синтез рапорту;

– прийняття рішення з видачею завдання;

– синтез послідовності дій;

– представлення результатів в технологічних формах звітності (наприклад, бланках).

2.4 Опис основних методів оптимізації режимів напруги в розподільній мережі

Для того щоб оцінити втрати напруги і розрахувати допустимі відхилення вузла, або робити налаштування регулюючих пристроїв, використовують розрахункові методи, які дають досить швидку оцінку стану мережі. Однак, є і недолік у даного методу, а саме: досить низька достовірність вихідної інформації, яку використовують під час розрахунків, призводить до низької точності результату розрахунків. В наслідок чого, метод не дозволяє з достатнім ступенем впевненості позначити діапазони відхилень.

Разом з тим, швидко розвиваються вимірювальні бази, і в першу чергу, сучасні засоби контролю, що дозволяють вирішувати розглянуті

завдання на основі інструментального підходу. При цьому можуть бути виміряні втрати та відхилення напруги в мережах, а це дозволить адекватно розробити і здійснити регулюючі заходи, враховуючи реальні закономірності зміни напружень в різних вузлах мереж.

Регулювання напруги за допомогою трансформаторів. Існує два основних способи регулювання напруги за допомогою трансформатора: РПН і ПБЗ.

Перемикання без збудження (ПБЗ) здійснюється трансформаторами, попередньо відключених від мережі. Їх виготовляють з 5 відгалуженнями (+ 2, + 1, 0, -1, -2). Відпайка (відгалуження) 0 вважається основним, а решта додатковими. На основному відгалуженні напруга дорівнює номінальному, і його коефіцієнт трансформації вважається номінальним. Коефіцієнти трансформації на додаткових відпайках відрізняються від номінального на -5, -2.5, +2.5 і +5 при їх використанні. Такі трансформатори не дозволяють забезпечити вимоги закону зустрічного регулювання.

Регулювання напруги під навантаженням (РПН) мають спеціальний автоматичний перемикаючий пристрій. Також регулювання під навантаженням дозволяє працювати з великим числом ступенів регулювальних відпайок і великим діапазоном для регулювання. РПН здійснюється перемиканням відпайок і зміною коефіцієнта трансформації під навантаженням, а отже, і забезпечує зустрічне регулювання протягом доби.

Для забезпечення необхідної якості електроенергії, пристрої РПН виготовлюються з автоматичними регуляторами напруги, до яких повинні пред'являтися високі вимоги, такі як: стійкість, достатня чутливість (без реакції на короткочасні відхилення напруг при КЗ, запусках потужних двигунів і т.д.), стабільність, легко регульовані температурні характеристики та інше.

3 ПРОЕКТУВАННЯ МОДЕЛІ СИСТЕМИ

Аналіз, існуючих в інформаційно-логічному напрямі підходів, свідчить про відсутність єдиної і закінченої концепції моделювання інтелектуальної діяльності людини у вигляді взаємодіючих логічних процесів. У зв'язку з цим, вироблено виділення наступних інформаційних процесів та запропонована їх структура. Для якісного проектування даної системи, потрібно виконати аналіз програмного забезпечення, порівняти існуючі алгоритми для вирішення даного завдання, і при необхідності, розробити власний на основі існуючих рішень. Також необхідно протестувати дану систему в наближених, до реальних, умовах.

3.1 Бази знань та моделі представлення знань

3.1.1 База знань

База знань – важлива компонента експертної системи, вона призначена для зберігання довгострокових даних, що описують розглянуту предметну область (а не поточних даних), і правил, що описують доцільні перетворення даних цієї області.

Як предметна область – обирається вузька (спеціальна) прикладна область. Далі, для створення ЕС в обраній області, збираються факти і правила, що поміщають в базу знань разом з механізмами виведення і спрощення. На відміну від всіх інших компонент ЕС, база знань – «змінна» частина системи, яка може поповнюватися і модифікуватися інженерами знань, і досвідом використання ЕС між консультаціями (а в деяких системах і в процесі консультації).

Існує кілька способів подання знань в ЕС, однак загальним для всіх них є те, що знання представлені в символьній формі (елементарними компонентами представлення знань є тексти, списки та інші символьні

структури). Тим самим, в ЕС реалізується принцип символьної природи міркувань, який полягає в тому, що процес міркування представляється у вигляді послідовності символьних перетворень.

Існують динамічні і статичні бази знань. Динамічна база знань змінюється з часом. Її вміст залежить і від навколишнього стану. Нові факти, що додаються в базу знань, є результатом виведення, що використовується в застосуванні правил до наявних фактів. У системах з монотонним висновком, факти, що збережені в базі знань – статичні, тобто не змінюються в процесі виконання завдання. У системах з немонотонним висновком допускається зміна або видалення фактів з бази знань.

Однією з найбільш важливих проблем, характерних для систем, заснованих на знаннях – є проблема подання знань. Це пояснюється тим, що форма подання знань має суттєвий вплив на характеристики і властивості системи. Для того щоб маніпулювати всілякими знаннями з реального світу за допомогою комп'ютера, необхідно здійснювати їх моделювання. У таких випадках необхідно відрізняти знання, призначені для обробки комп'ютером, від знань, які використовуються людиною.

При проектуванні моделі подання знань слід враховувати такі фактори, як однорідність подання і простота розуміння. Однорідне уявлення призводить до спрощення механізму керування логічним висновком та спрощення управління знаннями. Подання знань має бути зрозумілим експертам і користувачам системи. Однак виконати цю вимогу в рівній мірі, як для простих, так і для складних завдань досить важко.

Зазвичай, для нескладних завдань зупиняються на деякому середньому (компромісному) поданні, але для вирішення складних і великих завдань необхідне структурування і модульне подання.

3.1.2 Модель представлення знань

В даний час розроблені десятки моделей (або мов) подання знань для різних предметних областей. Класифікація моделей подання знань представлено на рисунку 3.1. Більшість з них може бути зведене до наступних класів:

– продукційні моделі;

– семантичні мережі;

– фрейми;

– формальні логічні моделі.

У свою чергу цю безліч класів можна розбити на дві великі групи:

– модульні;

– мережеві.

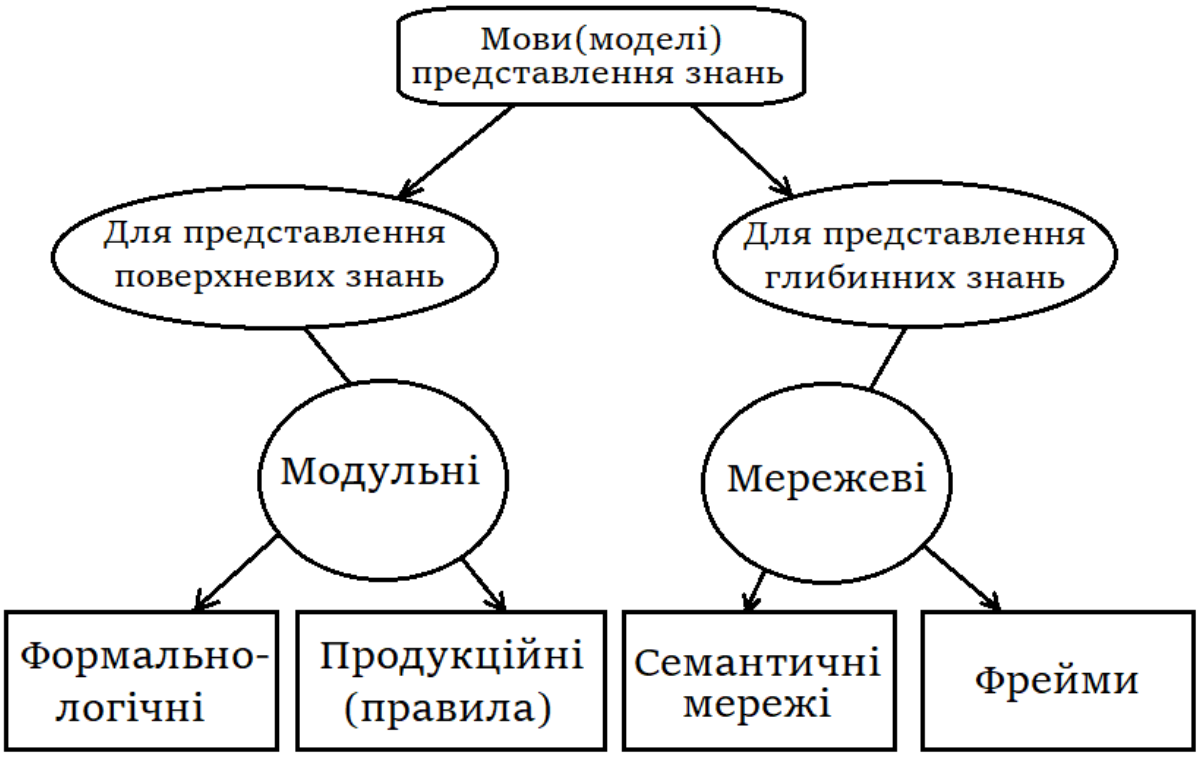

Рисунок 3.1 – Класифікація моделей подання знань

Модульні мови оперують окремими (не пов'язаними) елементами знань, будь то правила або аксіоми предметної області.

Мережеві мови надають можливість пов'язувати ці елементи або фрагменти знань через відносини в семантичні мережі або мережі фреймів.

Розглянемо докладніше найбільш популярні у розробників мови представлення знань (МПЗ).

3.1.2.1 Продукційна модель

Мови представлення знань, засновані на правилах (rule-based), є найбільш поширеними і більше 80% ЕС використовують саме їх.

Продукційна модель, або модель заснована на правилах, дозволяє уявити знання у вигляді пропозицій типу «Якщо (умова), то (дія)».

Під «умовою» (антецедентом) розуміється деякий пропозицію-зразок, за яким здійснюється пошук в базі знань, а під «дією» (консеквентом) - дії, що виконуються при успішному результаті пошуку (вони можуть бути проміжними, які виступають далі як умови, і термінальними або цільовими, завершальними для роботи системи).

Найчастіше висновок на такій базі знань буває прямий (від даних до пошуку мети) або зворотний (від мети для її підтвердження – до даних). Дані – це вихідні факти, що зберігаються в базі фактів, на підставі яких запускається машина виведення або інтерпретатор правил, перебираючий правила з продукційної бази знань.

Продукційна модель так часто застосовується в промислових експертних системах, оскільки привертає розробників своєї наочністю, високою модульністю, легкістю внесення доповнень і змін, простотою механізму логічного висновку, рисунок 3.2.

Є велика кількість програмних засобів, що реалізують продукційний підхід (наприклад, мови високого рівня CLIPS і OPS 5; «оболонки» або «порожні» ЕС – EXSYS і Карра, інструментальні системи KEE, ARTS, PIES.

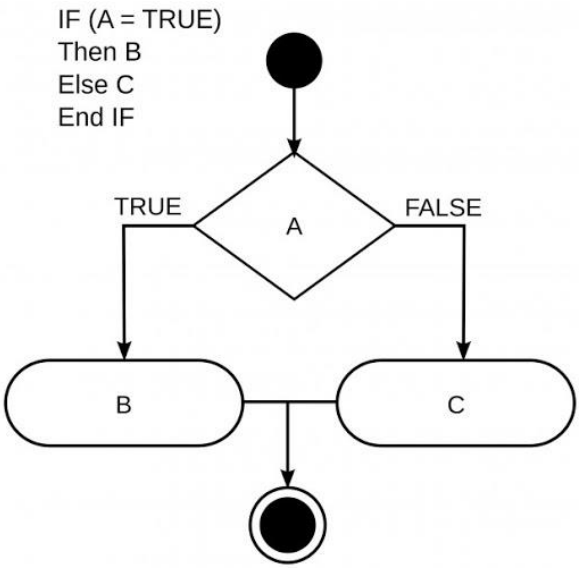

Рисунок 3.2 – Приклад продукційної моделі

Математичне подання продукційної моделі:

Психологічні дослідження процесів прийняття рішень людиною показали, що розмірковуючи і приймаючи рішення, людина використовує правила продукцій, або продукційні правила. У загальному випадку продукційне правило можна представити в наступному вигляді:

$$i: S; L; A \rightarrow B; Q \qquad (2.2)$$

де i – індивідуальний номер продукції;

S – опис класу ситуацій, в якому дана структура може використовуватися;

L – умова, за якою продукція активізується;

A→B – ядро продукції, наприклад: «Якщо A1, A2, ..., Ап то B». Такая запись означает, что «если все условия от A1 до An являются истиной, то B также истина» или же «когда все условия от A1 до An становятся истиной, то следует выполнить действие B»;

Q – постумова продукційного правила, описує операції і дії (процедури), які необхідно виконати після виконання B. Наприклад, внести зміни в дані або в саму продукцію.

Залежно від кількості умов і дій у відповідних переліках, розрізняють наступні типи правил: просте – одна умова і одна дія, складене – багато умов і дій, фіксуюче – багато умов і одна дія, розгалужене – одна умова і багато дій.

У загальному випадку, під умовою розуміється деяку пропозицію – зразок, за яким здійснюють пошук в базі знань, а під дією – дії, що виконуються при успішному результаті пошуку. Це можуть бути реальні дії, якщо система керуюча, або висновок, що представляє собою нове знання, або деяку мету.

3.1.2.2 Семантичні мережі

Термін «семантичного» означає «смислова», а сама семантика – це наука, що встановлює відношення між символами і об'єктами, які вони означають, тобто наука, яка визначає зміст знаків. Модель на основі семантичної мережі, що зображена на рисунку 3.3, була запропонована американським психологом Куілл-ном. Основною її перевагою є те, що вона більше за інших відповідає сучасним уявленням про організацію довготривалої пам'яті людини.

Семантична мережа – це орієнтований граф, вершини якого – поняття, а дуги – відношення між ними.

Як поняття зазвичай виступають абстрактні або конкретні об'єкти, а відносини це зв'язки типу: «це» (АКО – «A-Kind-Of», або «is» елемент класу), «має частиною» («has part»), «належить».

Можна запропонувати кілька класифікацій семантичних мереж, пов'язаних з типами відносин між поняттями.

За кількістю типів відносин:

– однорідні (з єдиним типом відносин);

– неоднорідні (з різними типами відносин).

По типам відносин:

– бінарні (в яких відносини пов'язують два об'єкти);

– n-арні (в яких є спеціальні відносини, що зв'язують більше двох понять).

Найбільш часто в семантичних мережах використовуються наступні відносини:

– елемент класу (троянда це квітка);

– атрибутивні зв'язки / мати властивість (пам'ять має властивість – обсяг);

– значення властивості (колір має значення – жовтий);

– приклад об’єкта класу (троянда, наприклад – чайна);

– зв'язок типу «частина-ціле» (велосипед включає кермо);

– функціональні зв'язки (визначені зазвичай дієсловами «виробляє», «впливає» …);

– кількісні (більше, менше, дорівнює ...);

– просторові (далеко від, близько від, за, під, над ...);

– тимчасові (раніше, пізніше, протягом ...);

– логічні зв'язки (і, або, не) та інші.

– Мінімальний склад відносин в семантичній мережі такий:

– об’єкт класу або АКО;

– атрибутивні зв'язки / мати властивість;

– значення властивості.

Недоліком цієї моделі є складність організації процедури виведення на семантичній мережі.

Ця проблема зводиться до нетривіального завдання пошуку фрагмента мережі, що відповідає деякій підмережі, яка відображає необхідний запит до бази.

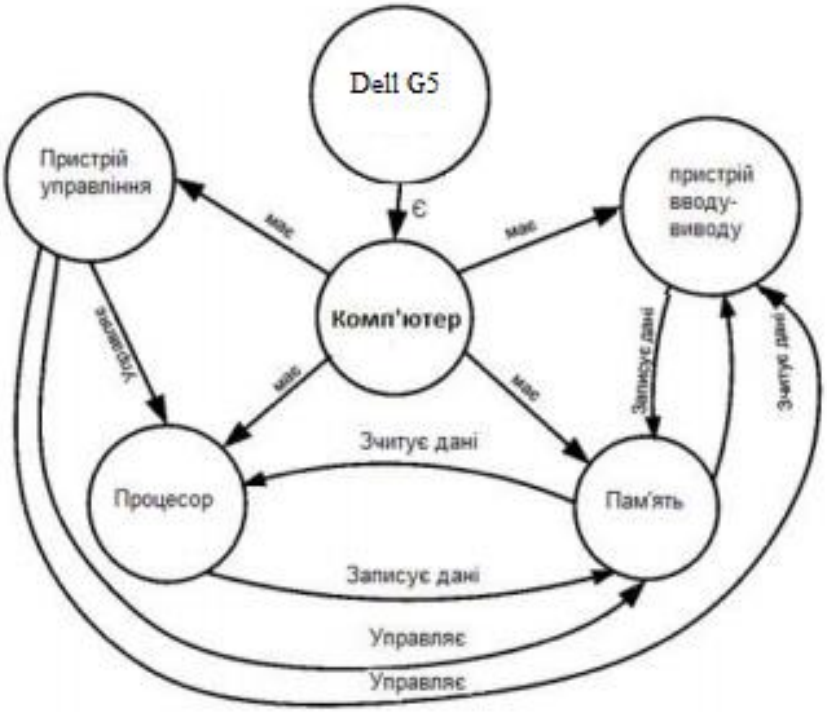

Рисунок 3.3 – Приклад семантичної мережі

3.1.2.3. Фрейми

Термін фрейм (від англ. Frame – «каркас» або «рамка») був запропонований Марвіном Мінським для позначення структури знань для сприйняття просторових сцен. Ця модель, як і семантична мережа, має глибоке психологічне обґрунтування.

Фрейм – це абстрактний образ для представлення стереотипу об'єкта, поняття чи ситуації. Структуру фрейму зображено на рисунку 3.4.

Інтуїтивно зрозуміло, що під абстрактним розуміється деяка узагальнена і спрощена модель або структура. Наприклад, проголошення вголос слова «кімната» породжує у слухачів образ кімнати: «житлове приміщення з чотирма стінами, підлогою, стелею, вікнами та дверима, площею 6-20 м2». З цього опису нічого не можна прибрати (наприклад, прибравши вікна, ми отримаємо вже комору, а не кімнату), але в ньому є «дірки» або «слоти» – це незаповнені значення деяких атрибутів,

наприклад: кількість вікон, колір стін, висота стелі, покриття підлоги та інші.

В теорії фреймів такий образ кімнати називається фреймом кімнати. Фреймом також називається і формалізована модель для відображення образу.

Розрізняють фрейми-зразки або прототипи, що зберігаються в базі знань, і фрейми-екземпляри, які створюються для відображення реальних фактичних ситуацій на основі даних, що надходять. Модель фрейма є досить універсальною, оскільки дозволяє відобразити все різноманіття знань про світ через:

– фрейми-структури, що використовуються для позначення об'єктів і понять (позика, застава, вексель);

– фрейми-ролі (менеджер, касир, клієнт);

– фрейми-сценарії (банкрутство, збори акціонерів, святкування іменин);

– фрейми-ситуації (тривога, аварія, робочий режим пристрою) та інші. Традиційно структура фрейму може бути представлена як список властивостей:

(Ім'я фрейма:

(Ім'я 1-го слота: значення 1-го слота), (ім'я 2-го слота: значення 2-го слота),

.....

(Ім'я N-го слота: значення N-го слота)).

Ту ж запис можна представити у вигляді таблиці, що представлена на рисунку 3.4, доповнивши її двома стовпцями.

У таблиці, додаткові стовпці (3-й і 4-й) призначені для опису способу отримання слотом його значення і можливого приєднання до того чи іншого слоту спеціальних процедур, що допускається в теорії фреймів. Як значення слота може виступати ім'я іншої фрейма, так утворюються мережі фреймів. Мережу фреймів зображено на рисунку 3.5.

Им'я фрейму			
Назва слоту	Значення слоту	Спосіб отримання значення	Приєднана процедура

Рисунок 3.4 – Структура фрейма

Існує кілька способів отримання слотом значень у фреймі-екземплярі:

– за замовчуванням від фрейму-зразка (Default-значення);

– через успадкування властивостей від фрейму, зазначеного у слоті АКО;

– за формулою, зазначеною в слоті;

– через приєднану процедуру;

– явно, по діалогу з користувачем;

– з бази даних.

Найважливішою властивістю теорії фреймів є запозичення з теорії семантичних мереж, так зване успадкування властивостей. І у фреймах, і в семантичних мережах спадкування відбувається за АКО-зв'язками. Слот АКО вказує на фрейм більш високого рівня ієрархії, звідки неявно успадковуються, тобто переносяться значення аналогічних слотів.

Основною перевагою фреймів як моделі подання знань є те, що вона відображає концептуальну основу організації пам'яті людини, а також її гнучкість і наочність.

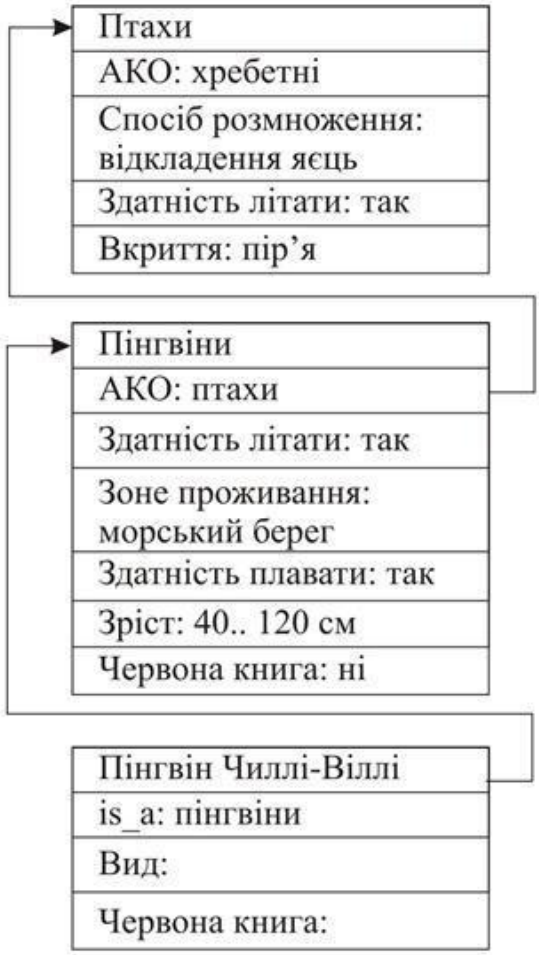

Рисунок 3.5 – Приклад фреймової мережі

3.1.2.4 Формальні логічні моделі

Традиційно в поданні знань виділяють формальні логічні моделі, засновані на класичному обчисленні предикатів 1-го порядку, коли предметна область або завдання описується у вигляді набору аксіом.

Реально обчислення предикатів 1-го порядку в промислових експертних системах практично не використовується. Ця логічна модель може бути застосована в основному в дослідницьких «іграшкових» системах, тому що пред'являє дуже високі вимоги і обмеження до предметної області. У промислових ж експертних системах використовуються різні її модифікації і розширення, зміст яких виходить за рамки необхідної системи.

3.2 Аналіз та вибір системи управління базою даних експертної системи діагностики.

При виборі СУБД необхідно враховувати безліч аспектів, кожен з яких накладає певні обмеження. Це не тільки технічні показники, а й економічні міркування, облік ринку.

База даних – сукупність даних, що організовані у відповідності до концептуальної структури, яка описує характеристики цих даних та відношення між ними, при чому така сукупність даних, яка підтримує одну або більше областей використання. У джерелах наводяться класифікації за декількома ознаками. За ступенем розподіленості існують централізовані та розподілені БД.

Централізовані – повністю підтримуються на одній ЕОМ. Розподіленими є ті, компоненти що розміщуються в різних вузлах комп'ютерної мережі у відповідності до певного критерію. Очевидно, що розподілені БД є більш надійними, оскільки при відмові одного з вузлів у мережі – інші залишаються дієздатними, а при умові, що кожен вузол буде мати резервні копії даних з інших вузлів, система буде мати високий рівень відмовостійкості. Проте, такі БД потребують значно більших ресурсів для підтримки роботи, а в контексті діяльності експертної системи, потрібно враховувати обмеженість ресурсів, як людських, так і матеріальних.

Також при взаємодії компонентів через мережу можна очікувати меншу швидкодію, порівняно з централізованою системою, що негативно відображається на швидкості роботи алгоритму, в якому навіть мілісекунди затримки при виникненні НЗ можуть бути життєво важливими. Саме тому для вирішення даної задачі можна зупинитися на централізованому типі БД, що значно спростить підтримку та використання. Така БД зможе бути використана або на сервері підстанції, або безпосередньо встановлена на ЕОМ диспетчера, відповідального за роботу ЕС та документацію, що генерується автоматизованою системою, яка розробляється. А надійність

бази даних можна забезпечити регулярним створенням резервних копій – дампів, хоча при використанні СУБД у складі експертної системи, така необхідність залежить від типу діяльності, що експертна система буде автоматизувати.

За середовищем постійного збереження відокремлюють БД, що зберігаються у вторинній, тобто постійній пам’яті комп’ютера (як правило жорсткий диск), в оперативній пам’яті та третинній – змінних носіях інформації. В даному випадку, оскільки для підтримки документації важливим фактором є надійність збереження даних, оптимальним варіантом слід вважати традиційну модель зберігання – у вторинній пам’яті. Вона програє у швидкодії оперативній пам’яті, проте менш вибаглива до характеристик ЕОМ, на якій буде працювати БД.

Більшість баз даних використовують мову SQL. Вона є стандартизованою, але різні реалізації в рамках певних СУБД можуть дещо відхилятися від цього стандарту з метою надання більших можливостей чи поліпшення швидкодії. Для використання в рамках даної роботи було прийнято рішення обрати реляційну базу даних, оскільки даний формат має велику розповсюдженість, відповідно наявна велика кількість джерел інформації стосовно використання таких БД, а також наявна велика кількість СУБД під цей тип, тобто є можливість обрати оптимальний варіант для використання в системі, що розробляється.

На сьогоднішній день існує досить багато систем управління базами даних. Особливої уваги заслуговує PostgreSQL. Безкоштовна об'єктно-реляційна система управління базами даних. Вона є однією з найбільш просунутих СУБД, що в першу чергу орієнтується на повну відповідність стандартам та можливості розширення, тобто повністю намагається відповідати SQL–стандартам ANSI/ISO. З-поміж інших, цю СУБД також відокремлює той факт, що вона має об’єктно-орієнтований функціонал та підтримує відповідні концепти.

Система PostgreSQL заснована на ядрі, створеному безліччю

розробників. У подібних випадках розумно зосередитися на оснащенні системи новими можливостями, але не займатися оптимальним їх втіленням, оскільки у випадку виникнення необхідності завжди можна буде повернутися до оптимізації відповідних ділянок коду. Також варто зазначити що ця система не достатньо оптимізована для вирішення повсякденних (не дуже важких) задач. Її використання більше передбачається у випадках, коли важливим виявляється підвищена надійність та підтримка об’єктних підходів до БД. У випадку, коли більш важливим є виконання простих операцій зчитування-запису, PostgreSQL показує не найкращі результати.

З характеристик СУБД, які можуть визначити вибір, однією з найважливіших є модель даних. Теоретично будь-яку інформацію можна представити у вигляді реляційної моделі. Сила реляційних баз даних в тому, що ця модель дуже добре підходить для підприємств, які мають у своєму розпорядженні чималі кошти для активного впровадження передових систем. Ця модель має найбільш відпрацьовану математичну підставу і добре опрацьовані стандарти. Реляційна модель даних відрізняється великою гнучкістю з точки зору зміни структури даних. Тут можна змінювати фізичну структуру даних, не переписуючи додатки.

Будучи СУБД класу підприємства, PostgreSQL надає такі особливості як Multi-Version Concurrency Control (MVCC), відновлення по точці в часі, табличний простір, асинхронна реплікація, вкладені транзакції (точки збереження), гаряче резервування, планувальник/оптимізатор запитів, і випереджаючий журнал на випадок поломки. Він підтримує міжнародні кодування, в тому числі і багатобайтові, при використанні різних кодувань можна використовувати сортування і повнотекстовий пошук, розрізняти регістр. Велика кількість підконтрольних даних і велике число одночасно працюючих користувачів, тим не менш, не сильно впливає на масштабованість системи. Є діючі PostgreSQL системи, які управляють більш ніж десятками терабайт даних.

PostgreSQL може виконувати збережені процедури, написані на різних мовах програмування, включаючи Java, Perl, Python, Ruby, Tcl, C / C++, і власному PL / pgSQL, аналогічному Oracle's PL / SQL. У стандартну бібліотеку функцій включені сотні вбудованих функцій – від базових математичних і строкових операцій до криптографічних функцій і функцій, що забезпечують сумісність з Oracle.

3.3 Вибір програмного забезпечення

Середовище розробки, в якій буде реалізована експертна система для діагностики аварійних ситуацій повинна володіти широким спектром можливостей і підтримувати такі технологій як:

– підтримка бази даних для накопичення інформації і формування більш об'єктивних перемикань у подальшому;

– кросплатформенність, що дозволяє спостерігати за роботою ЕС з будь-якого пристрою, особливо з мобільних;

– об'єктно-орієнтована мова програмування, що дозволяє робити більш якісну візуалізацію з широким функціоналом;

– реєстрація користувачів для підвищення безпеки;

– розмежовування права доступу до системи;

– облік (тривалості ремонтних робіт і частоти відмов електричних мереж);

– оптимізації, тому що сама програма повинна займати мінімальний обсяг оперативної пам'яті, для підвищення працездатності та часу виконання алгоритму.

У поточних реаліях існує величезна кількість програмного забезпечення для реалізації даного завдання. Наведемо опис деяких, найбільш близьких для виконання.

3.3.1 Unity 3D

Як програмне забезпечення у ролі інтерфейсу користувача було вибрано ПО Unity.

Unity – міжплатформленне середовище розробки комп'ютерних ігор, графічних додатків, що розроблена американською компанією Unity Technologies.

Перша перевага Unity – наявність кросплатформенного функціоналу, що дозволяє користувачам працювати в проекті з будь-якого пристрою. Другою гідністю є цінова політика і політика поширення двигуна системи – дуже приємна і гуманна: безкоштовна версія може розробляти комерційні додатки відразу під всі платформи, а професійна версія коштує всього 1500$ і надає додаткові функції, в першу чергу, що пов'язані з оптимізацією і графікою. Третій плюс платформи – інтуїтивно зрозумілий інтерфейс редактору і використання простих в освоєнні мов програмування: C# та JavaScript.

До недоліків відносять появу складнощів при роботі з багато-компонентними схемами і труднощі при підключенні зовнішніх бібліотек.

На Unity написані тисячі ігор, додатків, візуалізації математичних моделей, які охоплюють безліч платформ і жанрів.

Проект в Unity має рівні, які взаємодіють між собою шляхом приєднання скриптів на об'єкти. Скрипти підтримують мову програмування C#. Скрипти дозволяють привласнити функції певним об'єктам або цілому класу об'єктів, що полегшує розробку і забезпечує більш тонке налаштування. Також, за замовчуванням є компонент MeshRenderer, який робить модель об'єкта видимою. На рисунку 3.6 представлений інтерфейс ПО Unity.

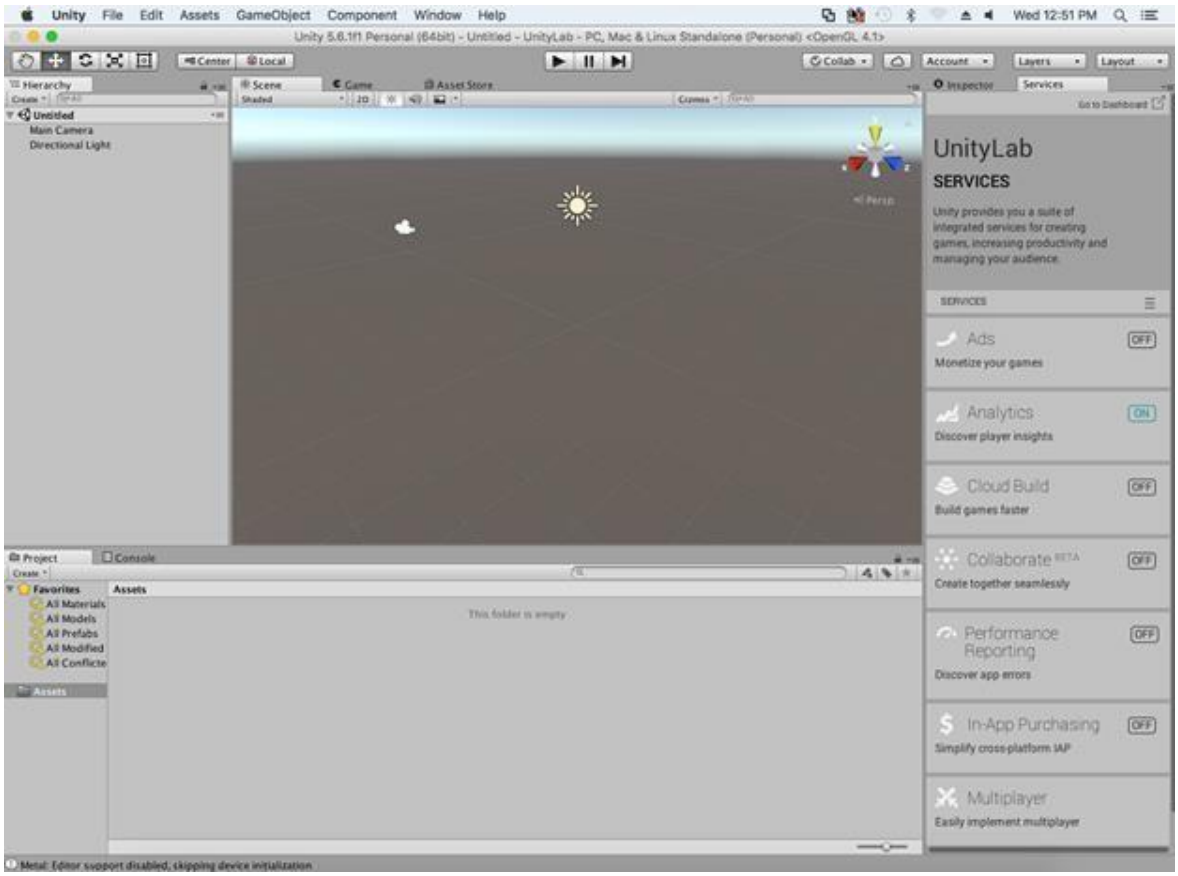

Рисунок 3.6 – Інтерфейс ПО Unity 3D

Підводячи підсумок огляду Unity, можна зробити висновок, що Unity – добрий вибір для реалізації середніх по складності проектів, як для мобільних пристроїв, так і для ПК. Однак, в рамках даного проекту Unity є надмірно «великим» додатком, так як необхідний функціонал буде використовувати можливості програми, тільки на половину.

3.3.2 IDE PhpStorm

PhpStorm – комерційне кросплатформенне інтегроване середовище розробки для PHP. PhpStorm розроблений на основі платформи IntelliJ IDEA, написаної на Java. Користувачі можуть розширити функціональність середовища розробки за рахунок завантаження плагінів, розроблених для платформи IntelliJ, або написавши власний плагін.

Середовище розробки, володіє наступними перевагами:

– підтримка SQL і баз даних;

– віддалене розгортання додатків;

– інтеграція з Git дозволяє працювати над проектом спільно та контролюючи версії;

– локальна історія;

– надання автодоповнення, перевірку стандартних тегів, властивостей, імен цілей і так далі;

– автодоповнення коду для JavaScript, HTML і CSS;

– підтримка HTML5;

– зміни в коді можна миттєво переглянути в браузері без перезавантаження сторінки.

По підтримці основних фреймворків, PhpStorm ідеально підходить для роботи з Symfony, Drupal, WordPress, Zend Framework, Laravel, Magento, Joomla!, CakePHP, Yii.

До недоліку даної IDE можна віднести відсутність безкоштовної версії, порівняно слабку україно/російськомовну інформативну базу по використанню фреймворків. На рисунку 3.7 представлений інтерфейс PhpStorm.

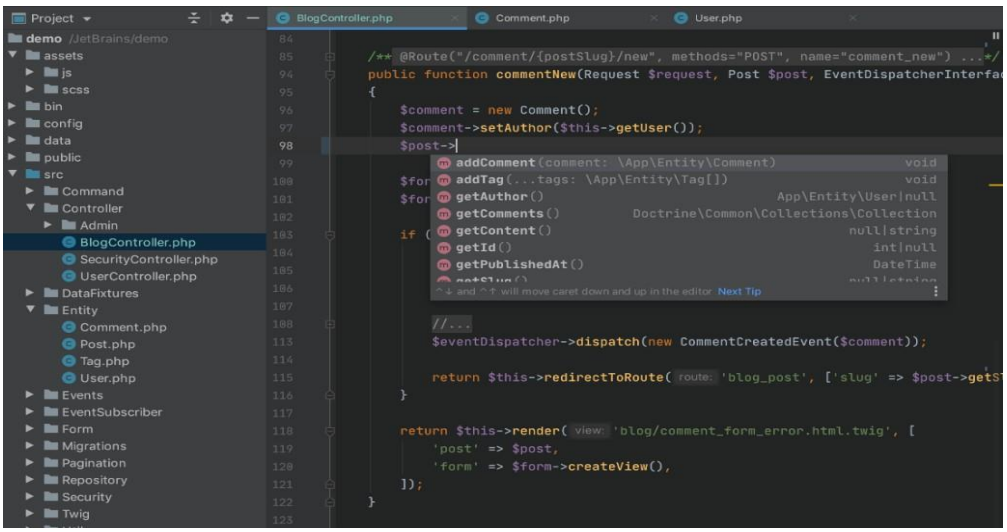

Рисунок 3.7 – Інтерфейс PhpStorm

У висновку огляду IDE PhpStorm хочеться відзначити велику варіативність в збірках. Користувач сам вирішує які плагіни і фреймворки використовувати, що дозволяє розробляти проекти від простого до складного.

3.4 Вибір алгоритму для обходу графа

Вибір алгоритму, для досягнення поставленої мети в магістерській роботі, є ключовим. Від правильно обраного алгоритму залежить швидкість роботи програми, правильність обраного маршруту, швидкість ремонтно-відновлювальних робіт.

Розподілена електрична мережа за своєю структурою, нагадує граф, приклад такої розподіленої мережі в Україні зображено на рисунку 3.8.

В наслідок чого, необхідно обрати алгоритм обходу графа. Існує безліч обходів графа: пошук в глибину, пошук в ширину, топологічне сортування і так далі. Однак, така велика безліч алгоритмів, не завжди відповідає необхідним вимогам.

Наприклад, більшість алгоритмів, проходять тільки по вузлах, які проходять по найкоротшому шляху. Деякі алгоритми, навпаки роблять надмірно багато заходів в вузли, щоб забезпечити входження в кожен вузол, працюють на неорієнтованому графі і більш того, більшість алгоритмів мають дуже велику обчислювальну складність, так як прораховують всі можливі маршрути до кожного з вузлів, що в промислових масштабах впливає на час усунення аварії.

Якщо адаптувати завдання даної роботи на графи, то необхідно розробити такий алгоритм, який буде в сукупності вирішувати відразу кілька завдань:

– забезпечення включень резервних фідерів найкоротшим способом;
– віддавати пріоритет вихідного шляху включення фідерів;
– чи враховує «закільцьованість»;

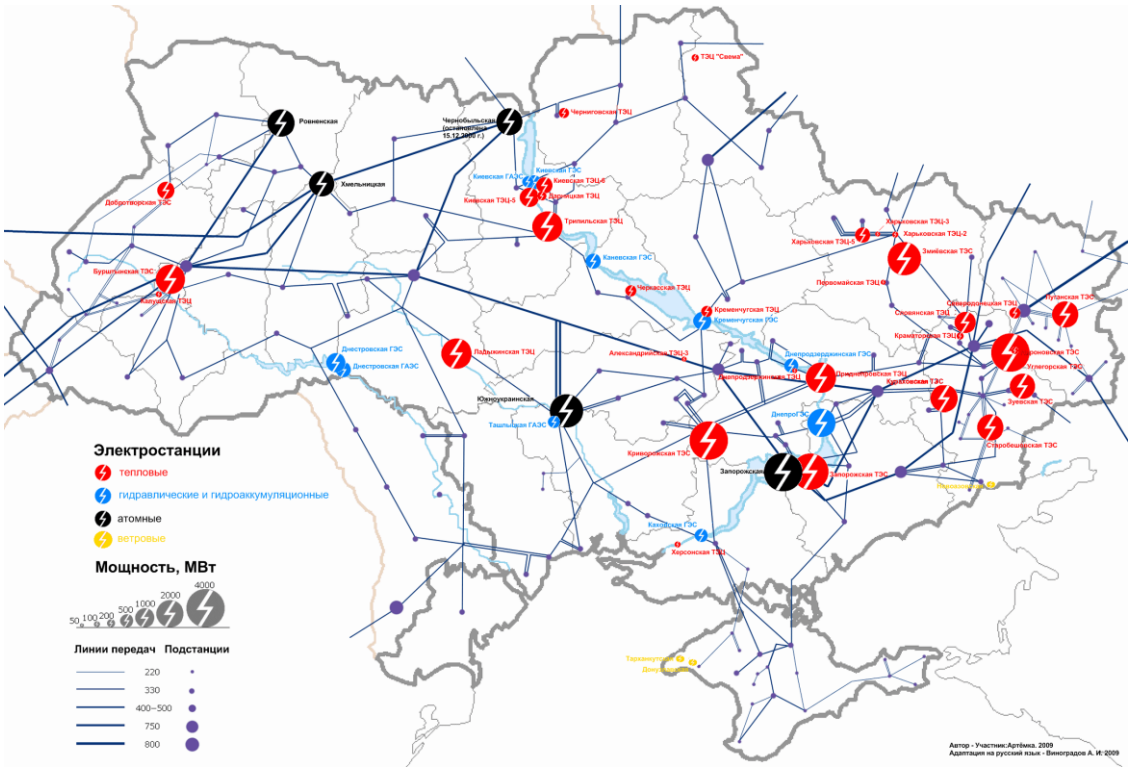

Рисунок 3.8 – Розподілена електрична мережа України

– враховувати ваги кожного фідера. Вагою в даній роботі виступає відстань від джерела електроенергії до підстанції. Надалі, при розвитку даної системи, можна домогтися більш тонкої конфігурації ваги, наприклад, враховувати навантаження на кожну підстанцію.

– включати резервний фідер, якщо можливо, паралельно, з метою підвищення надійності включення;

– виключити включення двох фідерів в одну підстанцію, так як це може привести до поломки самої електростанції;

– уміння прораховувати шляхи при примусовому зміну стану графа, без зміни всієї логіки роботи алгоритму. Алгоритм повинен бути достатньо гнучким;

– давати рекомендаційні поради диспетчеру;

– шукати шляхи з двох джерел електрики;

– можливість використання даної системи для подальшого проектування підстанцій.

3.4.1 Алгоритм Дейкстри

Алгоритм Дейкстри – це алгоритм графів, винайдений голландським вченим Едсгером Дейкстрою в 1959 році. Алгоритм виконує пошук найбільш коротшого шляху. Працює тільки для графів без ребер з негативним вагою.

Флагом (інцидентна трійка вершина-ребро-грань) в даному алгоритмі називається наявність або відсутність зв'язку між вузлами.

Принцип робота алгоритму полягає у тому, що всі вузли заповнені великим позитивним числом. Масив флагів – заповнений нулями. Потім починається основний цикл. Шукається вершина з мінімальною відстанню, і перевіряються всі вершини, що прилеглі до неї. Коли знаходиться найменша вершина, алгоритм переходить до неї і повторює свою роботу до тих пір, поки флаги всіх вершин не стануть рівними 1, або, коли всі вершини візьмуть значення 0 (даний випадок можливий тоді і тільки тоді, коли граф незв'язний).

Після того, як алгоритм проходить повну ітерацію вершин, він обчислює нову вагу для сусідніх вершин. Далі програма знаходить пік з найменшою вагою і на якому немає позначки, і повторює те ж саме. Реалізація програми припиняється, коли мітки розміщуються на всіх вершинах. Приклад алгоритм Дейкстри на графі зображено на рисунку 3.9.

Недоліком алгоритму є неможливість працювати з негативною вагою. Також алгоритм має велику складність роботи в виробничих масштабах.

Алгоритм Дейкстри і його реалізація.

Завдання. Нехай є країна G, в якій є безліч міст (позначимо цю безліч як V), і безліч доріг, що з'єднують пари міст (позначимо їх як E). Не факт, що кожна пара міст з'єднана дорогою. Іноді, щоб дістатися з одного міста в інший, слід відвідати кілька транзитних міст. У доріг є довжина. В країні G є місто-столиця s. Необхідно знайти найкоротші шляхи зі столиці до всіх інших міст.

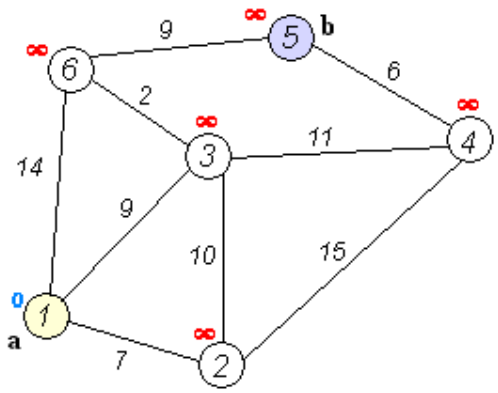

Рисунок 3.9 – Алгоритм Дейкстри на графі

Наведемо математичну формулювання цього завдання:

Нехай G = (V, E) –ні орієнтований граф, кожне ребро якого позначено невід'ємним числом (вага ребра). Помітимо деяку вершину s і назвемо її витоком. Знайти найкоротші шляхи від витоку s до всіх інших вершин графа G.

Це завдання має назву «Пошук найкоротшого шляху з одним джерелом».

Будь-яка частина найкоротшого шляху сама є найкоротший шлях. Це дозволяє для вирішення завдання застосувати динамічне програмування.

Метод динамічного програмування застосовується до завдань, які лежать в якомусь безлічі однотипних завдань, і деякі з завдань цієї безлічі прості і мають очевидні рішення, і крім того, є можливість знаходити рішення нових завдань ґрунтуючись на наявних рішеннях. Так в нашому випадку завдання «знайти найкоротший шлях з А в В» є одним із завдань виду «знайти найкоротший шлях з А в Х». Одна з цих завдань має очевидне рішення «знайти найкоротший шлях з А в А» – нікуди йти не потрібно, довжина найкоротшого шляху дорівнює 0. Залишилося винайти спосіб по-кроково розширювати безліч вирішених завдань зазначеного виду.

Про завдання, які вирішуються методом динамічного програмування, кажуть, що вони мають оптимальну підструктуру і

розпадаються на безліч перекриваючих підзадач. Оптимальність підструктури означає, що оптимальне рішення підзадач меншого розміру може бути використано для вирішення всього завдання у цілому. Як правило, при вирішенні таких завдань спочатку вибирається безліч параметрів, за якими ведеться обчислення. Зв'язок між параметрами і підзадачами встановлюється за допомогою рекурентних співвідношень. В процесі обробки підзадач використовується запам'ятовування їх проміжних результатів (зазвичай за допомогою масивів).

Наше завдання про найкоротший шлях володіє необхідною властивістю оптимальності для підзадач.

Нехай G = (V, E) – зважений орієнтований граф з ваговою функцією w: E → R +. Якщо p = (v1, v2, ..., vk) – найкоротший шлях з v1 у vk і 1 ≤ i ≤ j ≤ k, то pij = (vi, vi + 1, ..., vj) є найкоротший шлях з vi в vj.

Позначимо через d (u, v) величину найкоротшого шляху між вершинами u і v. Вага ребра між вершинами x та y будемо позначати w (x, y).

Слідство: розглянемо найкоротший шлях p з s в v. Нехай u → v – останнє ребро цього шляху. Тоді d (s, v) = d (s, u) + w (u, v).

Нехай G = (V, E) - зважений орієнтований граф з ваговою функцією w: E → R +. Нехай s Є V. Тоді для будь-якого ребра (u, v) Є E має місце нерівність:

$$d(s, v) \leq d(s, u) + w(u, v) \qquad (3.1)$$

Для вирішення завдання найкоротшого шляху з одним джерелом скористаємося жадібним алгоритмом Дейкстри. Принцип жадібного вибору тут виправданий, так як послідовність локально оптимальних виборів дає глобально оптимальне рішення. Формальне представлення алгоритму Дейкстри зображено на рисунку 3.10.

```
DIJKSTRA(G, w, s)
1  INITIALIZE_SINGLE_SOURCE(G, s)
2  S ← ∅
3  Q ← V[G]
4  while Q ≠ ∅
5        do u ← EXTRACT_MIN(Q)
6           S ← S ∪ {u}
7           for (для) каждой вершины v ∈ Adj[u]
8                 do RELAX(u, v, w)
```

Рисунок 3.10 – Формальне представлення алгоритму Дейкстри

Алгоритм Дейкстри будує множину S вершин, для яких найкоротші шляхи від витоку вже відомі. На кожному кроці до множини S додається та з вершин v, відстань до якої від витоку не більш, ніж до інших вершин. Таке додавання вершини v якраз і характеризує принцип жадібного вибору. Після додавання v в S найкоротша відстань від витоку до v вже ніколи не покращиться, встановимо used [v] = 1. Оскільки ваги дуг невід'ємні, то найкоротший шлях від витоку до конкретної вершини з безлічі S буде проходити тільки через вершини з S. Такий шлях будемо називати особливим. На кожному кроці алгоритму використовується масив d, в який записуються довжини найкоротших особливих шляхів для кожної вершини. Коли безліч S буде містити всі вершини графа (для всіх вершин будуть знайдені особливі шляху), тоді масив d буде містити довжини найкоротших шляхів від джерела до кожної вершини.

3.4.2 Гамільтонів шлях

Являє собою шлях в неорієнтованому або орієнтованому графі, який відвідує кожну вершину рівно один раз. Приклад гамільтонового шляху на графі зображено на рисунку 3.11.

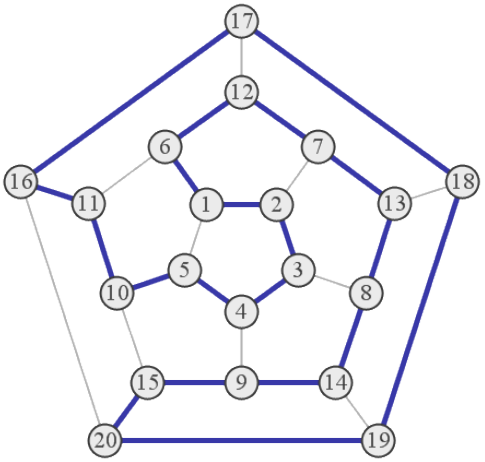

Рисунок 3.11 – Гамільтонів шлях на графі

Розглянемо цей алгоритм докладніше. Будемо вважати, що граф заданий околицями вершин: для кожної вершини X безліч вершин, суміжних з X. На кожному кроці алгоритму, є вже побудований відрізок шляху, він зберігається в стеці PATH. Для кожної вершини X, що входить в PATH, зберігається безліч всіх вершин N(X), суміжних з X, які ще не розглядалися в якості можливих продовжень шляху з вершини X. Коли вершина X додається до шляху, безліч N(X) вважається рівним. Надалі розглянуті вершини віддаляються з цієї безлічі. Черговий крок полягає в дослідженні околиці останньої вершини X шляху PATH. Якщо N(X)!=0 і в N(X) є вершини, які не належать шляху, то одна з таких вершин додається до шляху. В іншому випадку вершина виключається зі стека. Коли після додавання до шляху чергової вершини виявляється, що шлях містить всі вершини графа, залишається перевірити, суміжні чи перша і остання вершини шляху, і при позитивному відповіді видати черговий гамільтонів цикл.

Цей алгоритм дуже схожий на алгоритм пошуку в глибину і відрізняється від нього по суті тільки тим, що відкрита вершина, коли вся її околиця досліджена, не закривається, а знову стає новою (виключається зі стека). На початку всі вершини нові. Процес закінчується, коли всі вершини

знову стануть новими. Формальне представлення алгоритму проходження Гамільтонового шляху зображено на рисунку 3.12.

```
function findHamiltonianCycle(⟨V,E⟩):
      for v∈V:
        queue.pushBack(v)
      for k = 0..n * (n - 1)
        if (queue.at(0), queue.at(1)) ∉E
          i = 2
          while (queue.at(0), queue.at(i)) ∉E or (queue.at(1), queue.at(i + 1)) ∉E
              i++
        queue.swapSubQueue(1, i)
        queue.pushBack(queue.top())
        queue.pop()
```

Рисунок 3.12 – Формальне представлення алгоритму проходження Гамільтонового шляху

queue – черга вершин графа G=⟨V,E⟩.

Насправді це і є пошук в глибину, тільки не в самому графі, а в дереві шляхів. Вершинами цього дерева є всілякі прості шляхи, що починаються в вершині A, а ребро дерева з'єднує два шляхи, один з яких виходить з іншого додаванням однієї вершини в кінці. На малюнку 3.13 показаний граф і його дерево шляхів з вершини 1.

Даний алгоритм володіє якісною перевагою: а саме, дозволяє живити кожну підстанцію рівно один раз і виключає можливість «повторного входження в вузол».

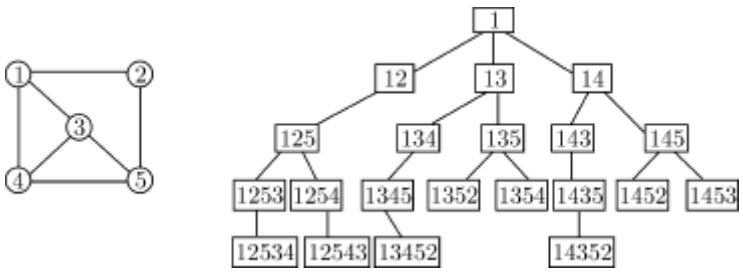

Рисунок 3.13 – Дерево шляхів з вершини 1

До недоліку даного алгоритму можна віднести відсутність роботи з зваженими графами, що робить даний алгоритм не актуальним для виробничих масштабів.

3.4.3 Покращений алгоритм Дейкстри

Даний розділ, пропонує покращений алгоритм, що поєднує в собі Гамільтонів шлях і алгоритм Дейкстри.

При виході з ладу підстанції / лінії електропередачі, першочергове завдання організації полягає у відновленні працездатності живлення всіх підстанцій в максимально короткі терміни. У зв'язку з цим, оптимальним буде маршрут з мінімальною кількістю перемикань, але за умови, що в роботу буде включено найбільшу кількість підстанцій.

Даний алгоритм буде відповідати всім вимогам для вирішення поставленого завдання, а саме:

– робота на зваженому графі;

– включення в роботу найбільшої кількості підстанцій;

– виключення можливості підключення двох фідерів в одну підстанцію;

– пріоритет віддається штатним шляхам живлення електричної мережі;

– не мати недоліків своїх попередників.

Мета даної модифікації полягає в підвищенні ефективності пропонованих перемикань, можливість працювати з виваженим графом і виключення помилок при пошуку альтернативного шляху, підвищенні якості функціонування алгоритму.

Цей результат досягається шляхом запам'ятовування всіх зв'язкових вузлів і прорахунку мінімальних маршрутів до всіх вершин, з урахуванням ліній, що знаходяться в стані аварії.

4 РОЗРОБКА АЛГОРИТМУ ТА ЙОГО РЕАЛІЗАЦІЯ

4.1 Функціональний аналіз мережі

Функціонально структурний аналіз ЕС, полягає в формуванні ланцюжка логічних міркувань, що призводять до однозначних висновків про стан мережі.

Ці міркування представляються у вигляді дерева, слідуючи по гілках якого, можна прийти до певної відповіді на питання про стан мережі.

Можна виділити чотири смислових і три функціональних варіанти комутаційної мережі, для представлення її як об'єкта з рівневої структурою.

Смислові варіанти:

– передача потоку за «нормально підключеною схемою».

– передача потоку за вимушеною схемою (відхилення від нормальної).

– не передача потоку до перевантажень.

– погашення частини мережі (зняття напруги).

Функціональні:

– здійснення передачі потоку енергії для всіх споживачів з необхідним рівнем надійності;

– передача потоку для всіх споживачів здійснюється при використанні резервних маршрутів;

– відсутність потоку енергії (виведення з роботи елементів без резервування).

Результатом роботи ЕС буде процес порівняння (нормального і передбачуваного з'єднання), агрегування (схеми мережі), класифікація (аварія або резервний шлях), складання рапорту.

Рапорт повинен відображати повний стан мережі і схему її комутації.

4.2 Розробка алгоритму та принцип його роботи

Для вирішення задачі пошуку маршруту, з мінімально можливою кількістю перемикань, був обраний «покращений алгоритм Дейкстри».

Даний алгоритм був обраний завдяки таким перевагам:

– робота на зваженому графі;

– включення в роботу найбільшої кількості підстанцій;

– виключення можливості підключення двох фідерів в одну підстанцію;

– пріоритет віддається штатному шляху живлення електричної мережі.

Нехай поточний стан енергомережі, представлений у вигляді графа, рисунок 4.1, у якому:

Ребра графів мають наступні атрибути:

– id ребра;

– статус ребра (активний / резервний / аварійний);

– пов'язані вузли (2 шт);

– протяжність лінії електропередач (вага ребра – int).

– вузли графів мають атрибути:

– id вузла;

– розрахункова вага мінімального маршруту (int);

– id вихідного вузла графа при розрахунку мінімального маршруту;

– статус (активний / неактивний (резервний) / аварійний).

У поточному стані всі вузли мають стан «активний».

Все фідери, пофарбовані в синій колір, мають стан – ввімкнений. Жовтий колір фідер – означає резервний маршрут. Червоний – аварія.

Початковий стан графа наведено на рисунку 4.1.

При виході з ладу однієї з магістралей, виконується наступний алгоритм:

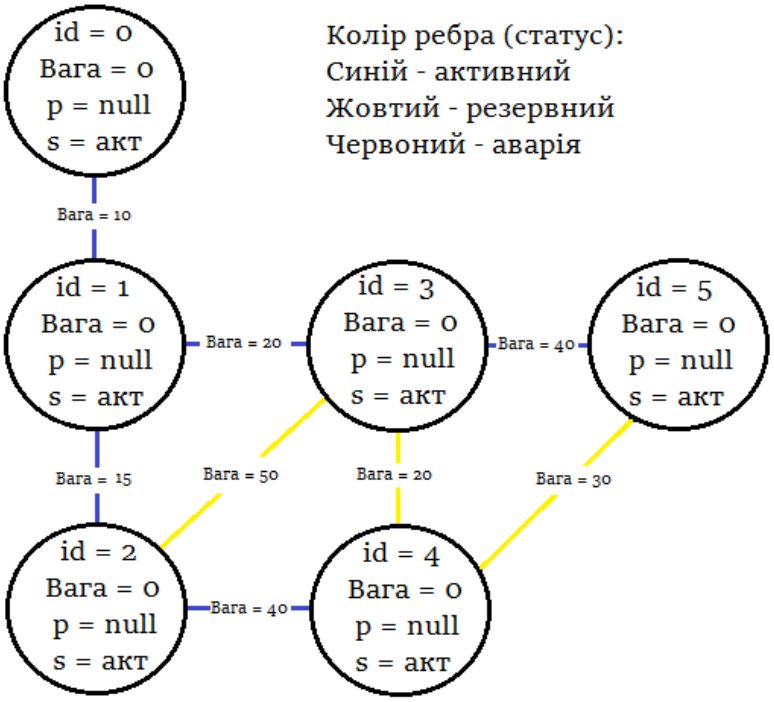

Рисунок 4.1 – Початковий стан системи

На рисунку 4.2 наведено перший крок роботи алгоритму – встановлюється статус обраного ребра як «аварійний».

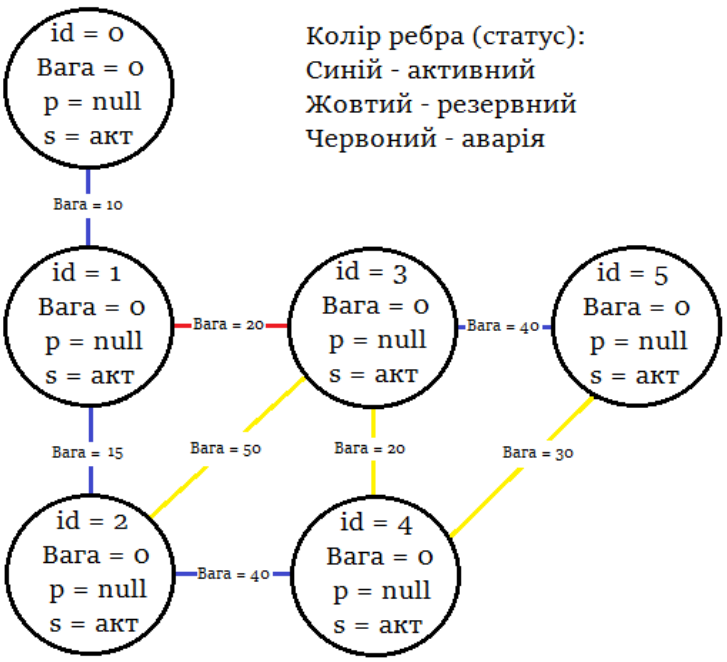

Рисунок 4.2 – Стан системи при аварії

На рисунку 4.3 показана конфігурація статусу всіх наступних за ним вузлів, пов'язаних активними ребрами, що не мають активних ребер, як аварійні.

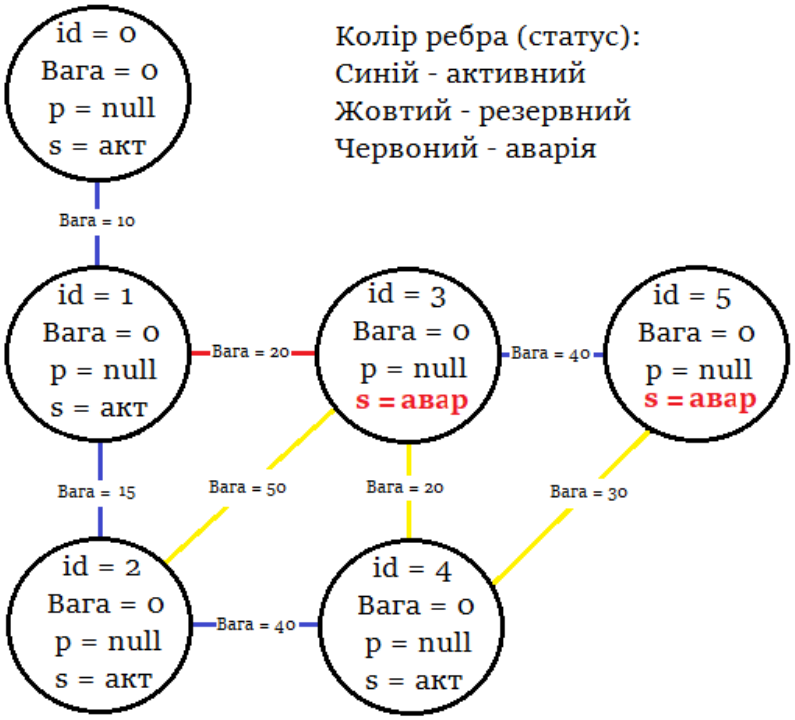

Рисунок 4.3 – Встановлення всіх пов'язаних підстанцій в стан «Аварії»

Основний алгоритм (модифікований алгоритм Дейкстри):

– встановлюємо: розрахункові ваги всіх вузлів в максимальне можливе значення. Вид, який набуває граф відображений на рисунку 4.4;

– id вихідного вузла графа при розрахунку мінімального маршруту = null;

– починаємо заповнювати розрахункові вузли графів і ID вихідних вузлів щодо початкового вузла (основної підстанції). При цьому використовуємо в якості ребер тільки активні.

Для цього, починаючи з початкового вузла визначаємо пов'язані з ним вузли. Якщо пов'язаний вузол має розрахункову вагу менше, ніж розрахункова вага поточного вузла + вага сполучного ребра, то для

пов'язаного вузла встановлюємо наступне:

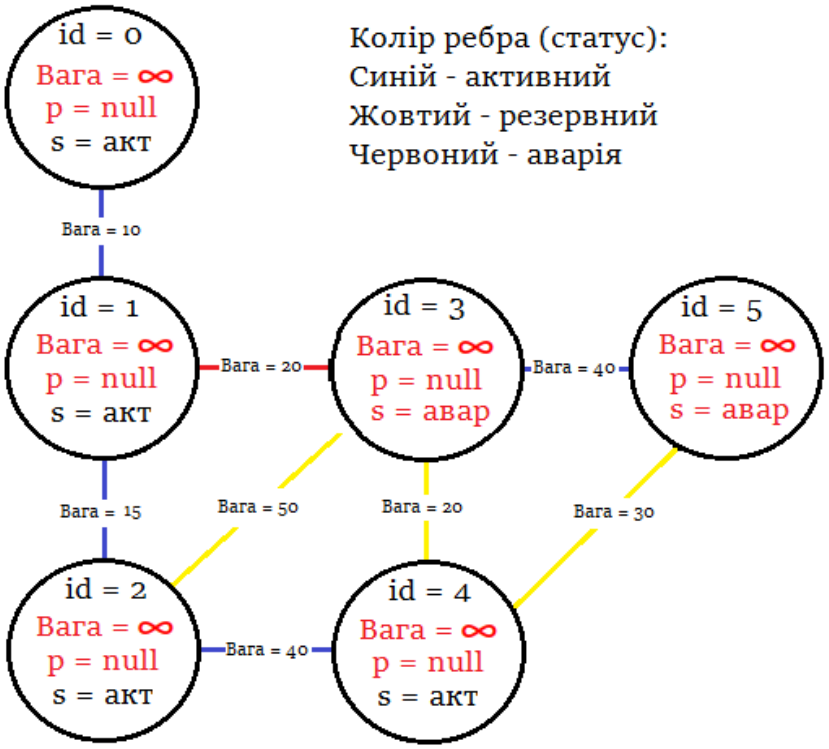

Рисунок 4.4 – Встановлення ваг і обнуління id вузлів

Розрахункова вага = розрахункова вага поточного вузла + вага пов’язаного ребра, а ID вихідного вузла необхідно встановити рівним ID поточного вузла. При рекурсивному заповненні у нас з’являється заповнений граф щодо аварійного стану. Стан системи, представлено на рисунку 4.5.

Далі алгоритм повинен «пройтися» по всіх вузлах, що мають стан «Активний». Продовжуємо заповнювати граф. Але тепер для резервних ребер.

Якщо пов'язаний вузол є аварійним, то обчислюємо розрахункову вагу. Якщо пов'язаний вузол має розрахункову вагу менше, ніж розрахункова вага поточного вузла + вага зв’язаного ребра, то для пов'язаного вузла встановлюємо: Розрахункова вага = розрахункова вага поточного вузла + вага пов'язаного ребра.

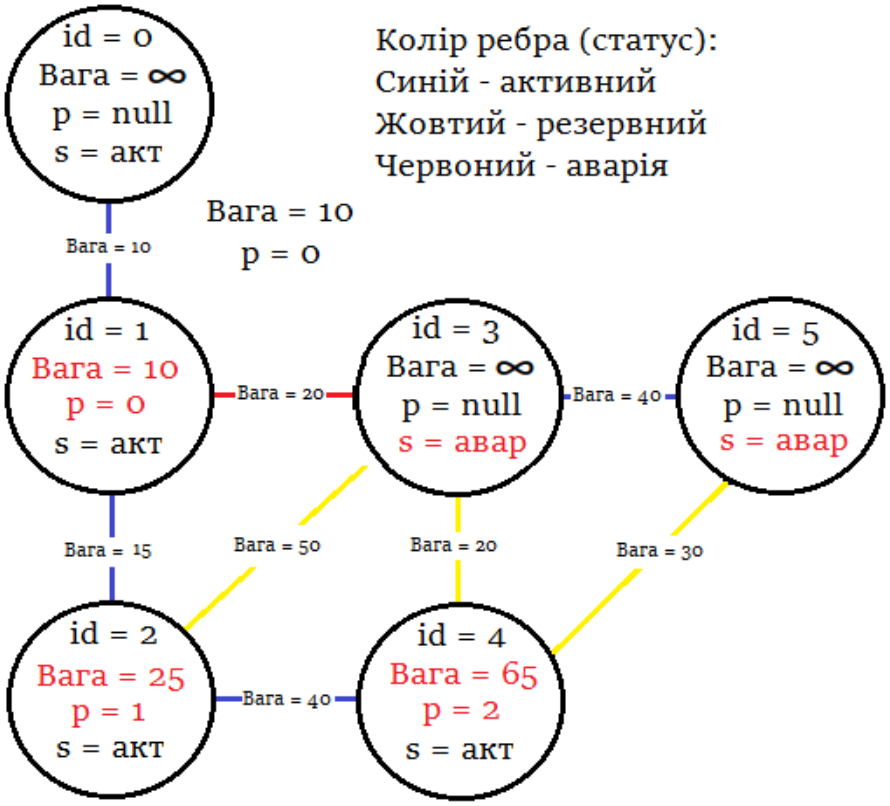

Рисунок 4.5 – Заповнення графа за допомогою розрахунку вузлів

ID вихідного вузла встановлюється рівним ID поточного вузла.

В даному випадку рекурсію проводити не слід. Результат даного кроку наведено на рисунку 4.6.

Наступним кроком, алгоритм повинен пройти по всім аварійним вузлам і знайти мінімальне значення розрахункової ваги (мінімального маршруту). Це і буде рішенням пошуку найкоротшого шляху (на рисунку 4.5 – це вузол з Id = 3). Для того, щоб визначити, які саме ребра слід включити, необхідно пройти по id вихідних вузлів графа від вузла, знайденого в попередньому пункті, до досягнення активного вузла.

Результат роботи алгоритму представлений на рисунку 4.7.

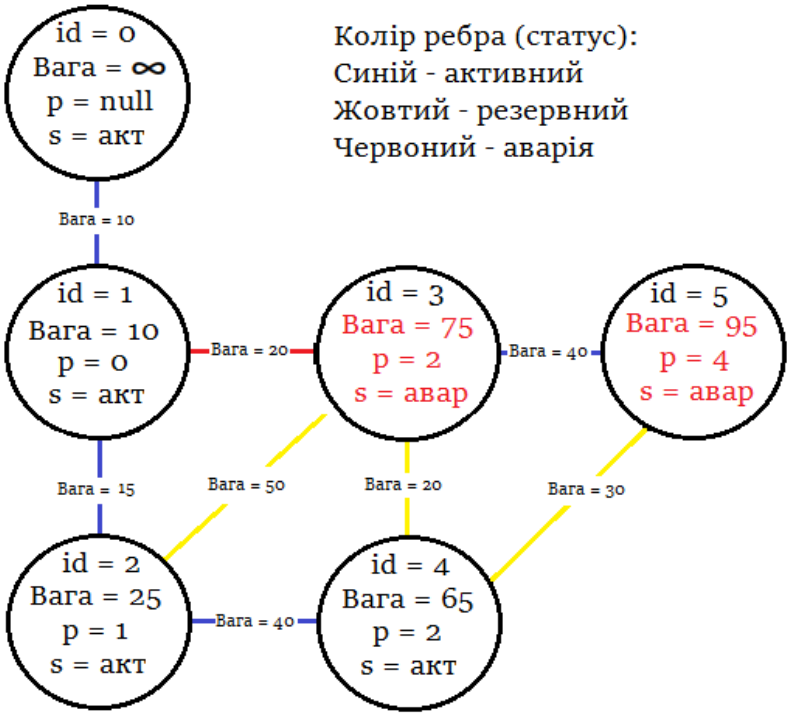

Рисунок 4.6 – Заповнення графа щодо резервних ребер

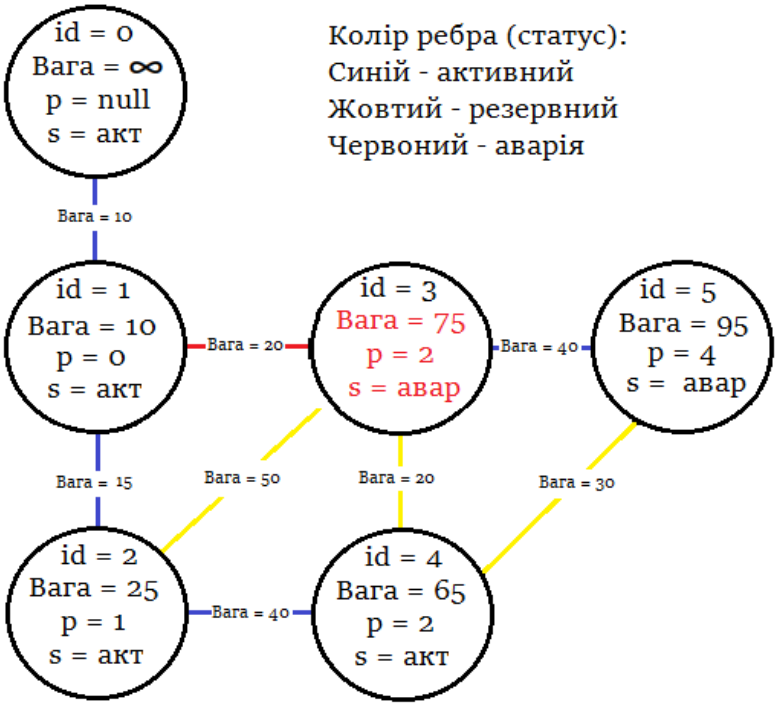

Рисунок 4.7 – Знайдений маршрут

4.3 Використовуваний стек технологій

Для реалізації вище запропонованого алгоритму, був обраний наступний програмний пакет:

PHP + Laraver + JS + PostgreSQL в середовищі розробки PhpStorm.

Вибір обґрунтований в основному за наявністю кросплатформенних складових (доступ з будь-якого пристрою) і можливістю підключення декількох користувачів. Більш того, є широкий спектр інструментів для розгортання даної технології в виробничих масштабах.

Короткий опис кожного компонента стека:

PHP – мова програмування. У роботі є інструментом для створення веб-сторінки.

Laravel + JS – за допомогою даного пакету, надаються можливості фреймворка: веб-додатки, в тому числі, тести, маршрутизація, перегляди та багато іншого. Лістинг даного стека, наведено в додатку Б.

PostgreSQL – база даних, що зберігає в собі інформацію про всі вузли і ребра, а також їх стан і статус.

4.4 Опис експертної системи

У даній роботі, вміст бази даних спроектована за допомогою SQL. Представлена база даних складається з 8 таблиць. Наведемо опис основних таблиць бази даних.

Таблиця «Edges»

На рисунку 4.8, представлена таблиця «зв'язків», дана таблиця є однією з ключових. Містить в собі інформацію про ваги кожного фідера, порядковий номер і зв'язку між підстанціями.

id	number	weight	linkedNode1_id	linkedNode2_id	edge_status_id
1	1	3	1	2	1
2	2	5	2	3	3
3	3	4	3	4	1
4	4	5	4	5	3
5	5	2	4	6	3
6	6	2	6	7	1
15	15	3	1	5	1
24	24	20	2	4	2
52	52	4	5	2	2
53	53	8	3	5	1
56	56	620	5	6	4

Рисунок 4.8 – Таблиця БД «Edges»

Number – порядковий номер фідера

Weight – вага фідера;

LinkedVertex1_id – початкова підстанція;

LinkedVertex2_id – кінцева підстанція;

Edge_status_id – статус фідера (аварійний, неактивний, активний, необхідно включити).

Таблиця «Vertices»

Дана таблиця містить інформацію про статус вузла, його положення, містить інформацію про джерела електроживлення і інформацію про попередній вузол (з ким пов'язаний). На рисунку 4.9 представлено наповнення даної таблиці.

id	number	weight	node_status_id	created_at	updated_at	x	y	primary	prev_node_id
1	1	(NULL)	1	2020-05-20 13:55:17	2020-06-05 16:23:36	0	-150	1	(NULL)
2	2	3	1	2020-05-20 13:55:17	2020-06-05 16:23:36	-200	-50	0	1
3	3	11	1	2020-05-20 13:55:17	2020-06-05 16:23:36	-200	150	0	5
4	4	15	1	2020-05-22 22:10:39	2020-06-05 16:23:36	100	150	0	3
5	5	3	1	2020-05-22 22:12:18	2020-06-05 16:23:36	100	-50	0	1
6	6	623	3	2020-05-23 14:56:40	2020-06-05 16:23:36	200	150	0	5
7	7	(NULL)	3	2020-05-23 14:57:46	2020-06-05 16:23:36	300	150	0	(NULL)

Рисунок 4.9 – Таблиця БД Vertices

Weight – величина, що показує пріоритет до перемикань (вираховується програмою).

Vertex_status_id – показує статус підстанції X-координата на X візуалізації Y-координата на У візуалізації.

Primary – показує джерело електроживлення.

Prev_vertex_id – попередня підстанція (вираховується алгоритмом, на основі даних з таблиці «Edges»).

Дана таблиця, має великі можливості для розширення спектру функціональності ЕС, наприклад, додавання другого джерела, опис підстанцій, адресація підстанцій до конкретного джерела живлення і багато іншого.

Решта таблиці, такі як «Vertex_status», «Edge_status» містять інформацію про статус підстанцій і фідерів, а також інформацію про кольори статусів, зміст таблиці «Edge_status» представлено на рисунку 4.10.

id	name	created_at	updated_at	color
1	Активен	2020-05-20 13:55:17	2020-05-20 13:55:17	green
2	Неактивен	2020-05-20 13:55:17	2020-05-20 13:55:17	orange
3	Аварийный	2020-05-20 13:55:17	2020-05-20 13:55:17	red
4	Необходимо включить	2020-05-22 14:38:00	2020-05-22 14:38:00	blue

Рисунок 4.10 – Зміст таблиці «Edge_status»

Існують порожні таблиці, такі як User, Reset_password, Failed_jobs створені для розширення функціоналу системи, що дозволяють додавати користувачів і давати їм права доступу, дивитися час і дати відмов, і час, витрачений на ремонт і так далі.

4.5 Тестування експертної системи

Представлені в системі методи і алгоритми, були застосовані в основі

програмного комплексу автоматизованого робочого місця (АРМ).

До складу АРМ увійшли:

– об'єктно-орієнтована база знань;

– редактори бази знань з графічним інтерфейсом, візуалізує зміст БД та інспектора комутаційного стану об'єктів і їх зв'язків;

– інтерпретатор завдань на перемикання;

– модулі аналізу комутаційного стану.

На рисунку 4.11 представлений інтерфейс ЕС і вихідна схема підключення, де: зелений колір – робочий стан, жовтий колір – резервне живлення, червоний колір – аварія, синій колір – необхідно ввімкнути.

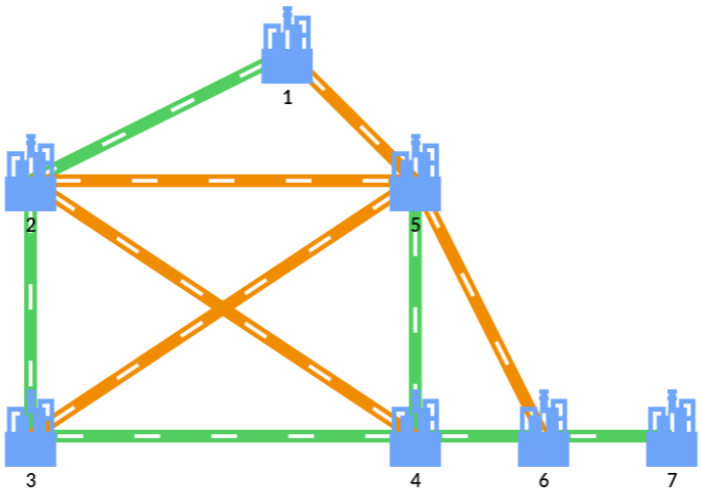

Рис. 4.11 – Початковий стан графа

Переведемо фідер 5-4 в аварійне становище. І натиснемо кнопку «Find_way».

Результатом виконання програми, повинен послужити рекомендація від ЕС – включити фідер 1-5 так як він має найменшу вагу, а отже, це найшвидше вирішення для забезпечення аварійного живлення підстанції №5. Результат роботи наведено на рисунку 4.12.

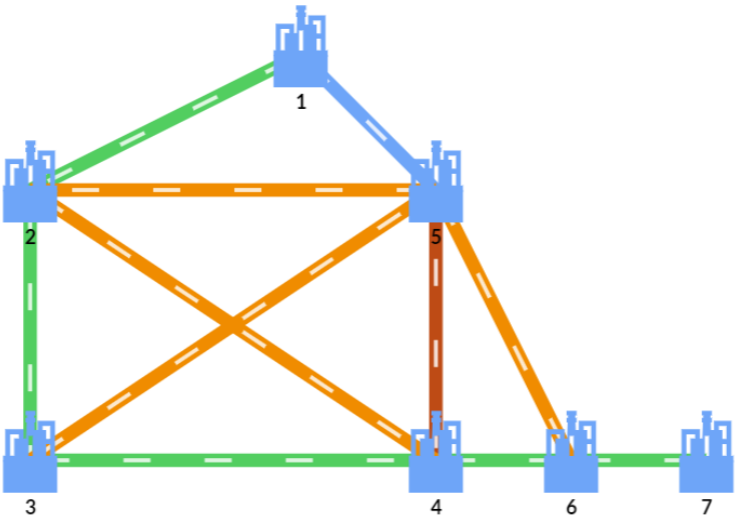

Рисунок 4.12 – Результат роботи ЕС при пошкодженні 4-5 фідера

Переведемо в режим «аварія» ще один фідер, наприклад, 3-4. Результатом роботи алгоритму, має послужити рекомендація включити 2-4 фідер. Результат наведено на рисунку 4.13.

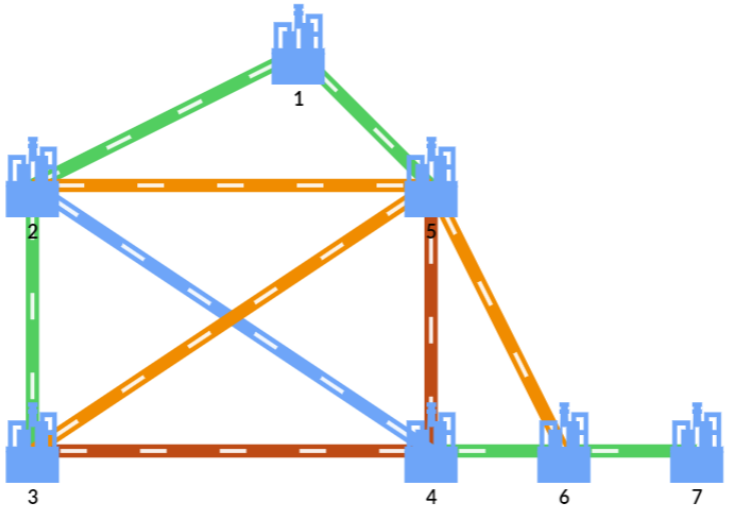

Рисунок 4.13 – Робота ЕС при двох аваріях

Більш того, система не просто змінює кольори фідерів і прораховує найкоротший шлях. ЕС змінює значення даної схеми комутації, що дозволяє використовувати змінені дані для подальшого аналізу або використання. На рисунку 4.14 показана змінена структура БД.

id	number	weight	linkedNode1_id	linkedNode2_id	edge_status_id
1	1	3	1	2	1
2	2	5	2	3	1
3	3	4	3	4	3
4	4	5	4	5	3
5	5	2	4	6	1
6	6	2	6	7	1
15	15	3	1	5	1
24	24	20	2	4	4
52	52	4	5	2	2
53	53	8	3	5	2
56	56	620	5	6	2

Рис. 4.14 – Зміни в БД

Як бачимо, алгоритм відпрацьовує справно, забезпечуючи мінімальну кількість перемикань і забезпечує ці перемикання найбільш ефективним чином.

ВИСНОВКИ

В даній кваліфікаційній роботі було показано, як використання такої технології як експертна система, заснована на знаннях, може оптимізувати роботу диспетчера електророзподільної станції.

Було проведено детальний аналіз існуючих інтелектуальних інформаційних систем у загальному плані та особливості функціоналу саме експертних систем заснованих на знаннях, для вирішення задач діагностики.

Було представлено структури взаємодіючих інтелектуальних процесів, що є конструктивною базою для розробки експертних систем та розподілу функцій між людиною та машиною. Перед розробкою системи було визначено функціональні вимоги до неї та обрано технології, за допомогою яких вона була реалізована.

У цій галузі проведено детальне порівняння алгоритмів, які дозволяють вирішувати завдання забезпечення напругою всіх абонентів електричної мережі на основі графа. Вибір зроблено на користь «Удосконаленого алгоритму Дейкстри», що дозволяє зробити мінімальну кількість перемикачів, з урахуванням найменшого шляху.

У рамках експертного підходу за допомогою програми розроблено та впроваджено наступні алгоритми:

– аналіз стану комутації мережі за допомогою процедур: перевірка підключеності, розподіл підключених частин, перевірка частоти навантаження джерел вузлів;

– агрегація;

– інтерпретації з процедурами: інтерпретація мережі для виділення її як об’єкта, інтерпретація завдань комутації, інтерпретація стану комутації;

– формування звіту про стан комутації мережі та синтез форм операційного перемикання з процедурами: пошук шунтуючого (обхідного) шляху для заданих ділянок, формування послідовності перемикання в первинній ціні.

На основі результатів магістерської роботи можна зробити висновок, що впровадження експертних систем на існуючій електростанції є досить пріоритетним та необхідним рішенням. Ця технологія більше підходить для вирішення проблем при проведенні капітального апгрейду технологічної та організаційної складової та на електророзподільній станції для тих підприємств та споживачів, які не можуть зупинити подачу електроенергії.

З введенням експертної системи в спільній роботі з диспетчером, час їх підключення та введення в експлуатацію значно зменшиться. Крім того, цю технологію можна використовувати на нових станціях, запланованих у майбутньому, для попереднього проектування місць для більш вигідного розташування електростанцій.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Березніков В.П. Організація обчислювального процесу при автоматизованій побудові діалогу в системі навчання операторів енергопідприємств. Дисертації: АН УРСР Київ, 1988. 20 с.

2. Філатов О.О. Перемикання в електричних розподільних пристроях. Москва: Енергія, 1985. 416 с.

3. Aiello N.A. Comparative study of control strategies for expert systems: AGE implementation of three variations of PUFF. / Aiello N. A. – In Proc. National Conference on Artificial Intelligence. 1986. С. 10–40.

4. Boose J. H. Expertise transfer and complex problems: using. AQUINAS as a workbench for knowledge-based systems./ Boose J. H., Bradshaw J. M. – International Journal of Man-Machine Studies, 26. 1987. С. 1–28.

5. Мандрикін С.О., Філатов О.О. Експлуатація та ремонт електрообладнання станцій та мереж. Москва: Вища школа, 1983. 344 с.

6. Самойлов В.Д., Березніков В.П., Писаренко А.П., Сметана С.І. Автоматизація побудови тренажерів і навчальних систем. Київ: Наукова Думка, 1989. 197 с.

7. Орнов В.Г., Рабінович М.О. Завдання оперативного і автоматизованого управління енергосистемами. Москва: Вища школа, 1988. 221 с.

8. Boose J.H. Gaines B. Knowledge Acquisition Tools for Expert Systems. New York: Academic Press, 1988. С. 7–65.

9. Buchanan B.G. Shortliffe E.H. Rule-Based Expert Systems. Reading MA: Addison-Wesley, 1984. С. 10–40.

10. Орнов В.Г., Любарський Ю.Я. Діалогові системи в диспетчерському управлінні енергооб'єктами. Москва: Вища школа, 1987. 151 с.

11. Орнов В.Г., Забігалов В.А., Семенов В.А. автоматизовані системи диспетчерського управління в енергосистемах. Москва: Вища школа, 1984. 264 с.

12. Любарський Ю.Я. Інтелектуальні інформаційні системи. Москва: Наука, 1990. 227 с.

13. Титоренко Г.А. Інформаційні технології управління. Москва: ЮНИТИ-ДАНА, 2003. 439 с.

14. Любарський Ю.Я., Кемельмахер Г.І. Застосування міні-ЕОМ в диспетчерському управлінні енергооб'єднання. Москва: ВІПК Енерго, 1984. 66 с.

15. Купершмідт Ю.Я., Любарський Ю.Я. Автоматизоване складання бланків перемикань енергооб'єктів. Москва: Електричні станції, 1984. №9, С. 8–80.

16. Lipmann R.P. Pattern classification using neural networks. IEEE Communications Magazine, 1989. С. 47–67.

17. Naylor C. Build your own expert system. Chichester: John Wiley&Sons Ltd, 1987. 289 с.

18. Бодянский Є.В. Руденко О.Г. Штучні нейронні мережі: архітектура, навчання. Харьков: ТЕЛЕТЕХ, 2004. — 372 с

19. Купершмідт Ю.Я., Любарський Ю.Я., Орнов В.Г. Принципи побудови універсального програмованого тренажера оперативних перемикань. Електричні станції, 1982. С. 48–52.

20. Суботін С.О. Подання та обробка знань у системах штучного інтелекту та підтримки прийняття рішень: Навчальний посібник. Запоріжжя: ЗНТУ, 2008. 341 с.

21. Стефік М., Ейкінс Я., Балзер Р. Організація експертних систем. Кібернетичний збірник, видання 22. Москва: Мир, 1985. С. 170–221.

22. Івашко В.Г., Фінн В.К. Експертні системи і деякі проблеми їх інтелектуалізації. Смуток і інформатика, видання 27. Москва: ВІНІТІ, 1986, С. 25–61.

23. Боуман Д., Емерсон С.Л., Дарновскі М. Практичний посібник SQL. Використання мови структурованих запитів. Москва: Вільямс, 2001. 336 с.

24. Сайт PostgreSQL. URL: https://www.postgresql.org/docs (дата звернення: 09.04.2021).

25. Сайт Borabai. Фізичне проектування бази даних. URL: https://bourabai.ru/dbt/dbms/03.html (дата звернення: 21.04.2021).

26. ДСТУ 3008: 2015. Документація. Звіти у сфері науки и техніки. Структура та правила оформлювання. Київ: ДП «УкрНДНЦ», 2016. 31 с.

27. Загальні методичні вказівки з дипломного проектування в університеті. Харків: ХНУРЕ, 2003. 40 с.

28. Методичні вказівки відносно структури, змісту та оформлення навчально-методичної літератури для авторів (упорядників) оригіналів. Харків: ХНУРЕ, 2002. 60 с.