

МАГІСТЕРСЬКА РОБОТА

МР. ШМ - 01.00.00.000 ПЗ

Група Шмз-24-1

Вовк Роман

2025

Івано-Франківський національний технічний університет нафти і газу

Інститут післядипломної освіти

Кафедра інженерії програмного забезпечення

Вовк Роман Богданович

(прізвище, ім'я, по батькові)

УДК 004.942
(індекс)

МАГІСТЕРСЬКА РОБОТА

Алгоритми, моделі та концепції штучного інтелекту для побудови

навчальних систем

(назва роботи)

Інженерія програмного забезпечення

(назва освітньої програми)

121 - Інженерія програмного забезпечення

(шифр і назва спеціальності)

Вовк Р.Б.

(підпис, ініціали та прізвище здобувача освітнього ступеня)

Науковий керівник **Шекета Василь Іванович, д.т.н., професор**

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Допущено до захисту

Завідувач кафедри

доц.

Бандура В.В.

(посада) (підпис) (дата) (ініціали та прізвище)

Нормоконтроль

асист.

Ваврик Т.О.

(посада) (підпис) (дата) (ініціали та прізвище)

Робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

Івано-Франківськ – 2025

Івано-Франківський національний технічний університет нафти і газу

Інститут післядипломної освіти

Кафедра інженерії програмного забезпечення

Освітній рівень магістр

Спеціальність 121 – Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ:

Зав. кафедрою

ІІЗ

доц.

В.В. Бандура

“ 04 ” вересня 2025 р.

ЗАВДАННЯ

НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ

Вовку Роману Богдановичу

(прізвище, ім'я, по-батькові)

1. Тема магістерської роботи “ Алгоритми, моделі та концепції штучного інтелекту для побудови навчальних систем ”

керівник проекту (роботи) Шекета Василь Іванович, д.т.н., професор

затверджені наказом закладу вищої освіти від “ 03 ” листопада 2025 р. № 198/12

2. Строк подання студентом проекту (роботи) 15 грудня 2025 р.

3. Вихідні дані до проекту (роботи) Теоретичні концепції та формальні моделі побудови інформаційних технологій функціонування навчальних систем на основі штучного інтелекту

4. Зміст розрахунково - пояснювальної записки(перелік питань, які потрібно розробити)

1. Аналіз концепцій штучного інтелекту для побудови навчальних систем
2. Дослідження методів концепції задоволення обмежень в для побудови навчальних систем
3. Формалізоване представлення концепції задоволення обмежень в термінах навчальних систем
4. Імплементация концепцій штучного інтелекту для побудови навчальної системи

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

1. Класифікація автоматизованих навчальних систем (рис. 1.1)
2. Структуризація концепцій ШІ для використання в навчальних системах (рис. 1.2)
3. Трьохкомпонентна модель контролю навчального процесу в АНС (рис. 1.3)
4. Модель онлайн навчання (рис. 2.1)
5. Графічне представлення задачі автоматичної генерації тестів (рис. 2.2)

6. Консультанти розділів проекту (роботи)

Розділ	Консультант	Підпис, дата

7. Дата видачі завдання 04 вересня 2025 р.

Керівник _____

(підпис)

Завдання прийняв до виконання _____

(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назви етапів магістерської роботи	Строк виконання етапів роботи	Примітка
1	Дослідження особливостей автоматизованих та інтелектуальних навчальних систем	15.09.2025	виконано
2	Аналіз концепцій штучного інтелекту для побудови навчальних систем	29.09.2025	виконано
3	Дослідження методів концепції задоволення обмежень в для побудови навчальних систем	15.10.2025	виконано
4	Формалізоване представлення концепції задоволення обмежень в термінах навчальних систем	08.11.2025	виконано
5	Імплементация концепцій штучного інтелекту для побудови навчальної системи	20.11.2025	виконано
6	Реалізація функціональності інтелектуальної навчальної системи	01.12.2025	виконано
7	Затвердження пояснювальної записки роботи завідувачем кафедри	15.12.2025	виконано

Студент – магістр _____

(підпис)

Керівник роботи _____

(підпис)

АНОТАЦІЯ

Магістерська робота: 93 с., 23 рис., 50 джерел.

Тема: Алгоритми, моделі та концепції штучного інтелекту для побудови навчальних систем

Об'єкт дослідження: автоматизовані та інтелектуальні навчальні системи, що використовують методи штучного інтелекту.

Мета роботи: розробка інтелектуальної навчальної системи на основі концепції задоволення обмежень (CSP) для підвищення ефективності процесу навчання та персоналізації освітнього контенту.

Предмет дослідження: методи та алгоритми концепції задоволення обмежень (CSP) у контексті побудови інтелектуальних навчальних систем.

Результати дослідження

В роботі запропоновано новий підхід до побудови інтелектуальних навчальних систем на основі концепції задоволення обмежень (CSP). Розроблено модель адаптивного персоналізованого навчання, що враховує когнітивні особливості студентів та динамічно змінює освітній контент.

Висновок

Запропонована інтелектуальна навчальна система поєднує методи експертних систем, ієрархічне моделювання навчального контенту та адаптивні підходи до навчання. Її особливості, зокрема підтримка навчання на основі множин обмежень та зворотного зв'язку, сприяють підвищенню ефективності освітнього процесу.

ШТУЧНИЙ ІНТЕЛЕКТ, ІНТЕЛЕКТУАЛЬНІ НАВЧАЛЬНІ СИСТЕМИ, АВТОМАТИЗОВАНЕ НАВЧАННЯ, КОНЦЕПЦІЯ ЗАДОВОЛЕННЯ ОБМЕЖЕНЬ (CSP), ПЕРСОНАЛІЗОВАНЕ НАВЧАННЯ, АДАПТИВНЕ ТЕСТУВАННЯ, МОДЕЛЬ НАВЧАННЯ.

ABSTRACT

Master Thesis: 93 pp., 23 fig., 50 sources.

Thesis Subject: Algorithms, models and concepts of artificial intelligence for building educational systems

Object of research: automated and intelligent educational systems using artificial intelligence methods.

Purpose of work: development of an intelligent educational system based on the concept of constraint satisfaction (CSP) to increase the efficiency of the learning process and personalization of educational content.

Subject of research: methods and algorithms of the concept of constraint satisfaction (CSP) in the context of building intelligent educational systems.

Research results

The paper proposes a new approach to building intelligent educational systems based on the concept of constraint satisfaction (CSP). A model of adaptive personalized learning has been developed that takes into account the cognitive characteristics of students and dynamically changes educational content.

Conclusion

The proposed intelligent educational system combines expert system methods, hierarchical modeling of educational content and adaptive approaches to learning. Its features, in particular, support for learning based on sets of constraints and feedback, contribute to increasing the efficiency of the educational process.

ARTIFICIAL INTELLIGENCE, INTELLIGENT LEARNING SYSTEMS, MACHINE LEARNING, CONCEPT OF CONSTRAINT SATISFACTION (CSP), PERSONALIZED LEARNING, ADAPTIVE TESTING, LEARNING MODEL.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ	9
ВСТУП.....	10
РОЗДІЛ 1. АНАЛІЗ КОНЦЕПЦІЙ ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ ПОБУДОВИ НАВЧАЛЬНИХ СИСТЕМ	14
1.1. Дослідження особливостей автоматизованих та інтелектуальних навчальних систем	14
1.3. Формування стратегії засвоєння знань у навчальних системах на основі штучного інтелекту.....	22
1.3.1. Біхевіористична теорія навчання у контексті автоматизованих навчальних систем.....	23
1.3.2. Використання навчальних ефектів в інтелектуальних АНС на основі концепції представлення та задоволення обмежень	24
1.3.3. Адаптація підходу Краудера в автоматизованих навчальних системах	26
1.4. Модель контролю навчального процесу.....	27
1.5. Опис мультимодальної моделі онлайн навчання	29
Висновки до розділу	32
РОЗДІЛ 2. ДОСЛІДЖЕННЯ МЕТОДІВ ТА АЛГОРИТМІВ КОНЦЕПЦІЇ ЗАДОВОЛЕННЯ ОБМЕЖЕНЬ (CSP) В КОНТЕКСТІ ПОБУДОВИ НАВЧАЛЬНИХ СИСТЕМ.....	34
2.1. Особливості концепції задоволення обмежень (CSP).....	34
2.2. Формалізоване представлення концепції задоволення обмежень в термінах навчальних систем	36
2.3. Використання концепції CSP для побудови навчальних систем та систем тестування знань.....	39
2.3.1. Автоматизоване складання розкладу.....	39

2.3.2. Задача персоналізованого навчання	40
2.3.3. Використання CSP у системах тестування знань.....	40
2.4. Застосування концепції CSP для задачі автоматичної генерації тестів	41
2.5. Програмна реалізація персоналізованого навчання на основі CSP.....	45
Висновки до розділу	49
РОЗДІЛ 3. ІМПЛЕМЕНТАЦІЯ КОНЦЕПЦІЙ ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ ПОБУДОВИ НАВЧАЛЬНОЇ СИСТЕМИ	51
3.1. Реалізація структурної схеми навчальної системи.....	51
3.2. Представлення моделей модулів інтелектуальної навчальної системи	54
3.2.1. Значення механізму інференції для ефективності навчальних систем.....	54
3.2.2. Класифікація користувачів та процес функціонування інтелектуальної навчальної системи	55
3.3. Представлення рівнів інтелектуальності пропонованої системи	57
3.4. Формалізоване представлення моделі об'єкта навчання в навчальній системі	59
3.4.1. Вимоги до реалізації інтерфейсу системи	61
3.4.2. Роль обмежень у навчальному процесі	62
3.5. Реалізація функціональності інтелектуальної навчальної системи	65
Висновки до розділу	76
ВИСНОВКИ	78
ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	80
ДОДАТКИ	85

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

АНС – автоматизована навчальна система

ІНС – інтелектуальна навчальна система

ІОТ - інтелектуальні освітні технології

ШІ – штучний інтелект

VR - віртуальна реальність

AR - доповнена реальність (AR)

AI - Artificial Intelligence

CSP - constraint satisfaction problem

BSP - Boolean Satisfiability Problem

ILP - Integer Linear Programming

MRV - Most Constrained Variable

IRT - Item Response Theory

ВСТУП

Актуальність теми.

Дослідники використовують штучний інтелект у галузі освіти для створення інтелектуальних систем навчання. Цей підхід спрямований на встановлення доступних методів навчання, які можна налаштувати та адаптувати до потреб та рівня знань окремих об'єктів навчання. Завдяки інтеграції штучного інтелекту в освіту дослідники прагнуть підвищити якість навчання, враховуючи різні уподобання об'єктів навчання. Крім того, ці інтелектуальні технології надають підтримку викладачам.

Інтелектуальні освітні технології (ІОТ) є одним з яскравих прикладів таких інновацій, заснованих на кількох теоретичних основах, що поєднують педагогічні принципи, психологічні концепції та методи штучного інтелекту. Індивідуалізація та диференціація навчання стали фундаментальними аспектами ІОТ. Ще однією ключовою теоретичною основою ІОТ є конструктивізм, який підкреслює активну взаємодію учнів з навчальним середовищем. Такі середовища створюються за допомогою різних передових технологій, таких як машинне навчання, аналіз даних, інтерактивні технології, обробка природної мови, розпізнавання мовлення та великі дані.

Теоретичні основи інтелектуальних освітніх технологій [1] підкреслюють важливість адаптації освіти до індивідуальних потреб, заохочення об'єктів навчання до навчання та включення постійного зворотного зв'язку та коригувань під час навчального процесу. Ці основні принципи слугують основою для створення та впровадження інструментів, які пропонують ефективний та індивідуальний навчальний досвід для кожного студента. Ключовою метою технології є покращення навчального процесу для кожного студента з урахуванням його характеристик та потреб. Це передбачає налаштування досвіду на основі особливостей кожного учня, включаючи темп засвоєння інформації, існуючий рівень знань, інтереси та бажані методи навчання. Персоналізовані підходи до навчання здатні

задовольнити індивідуальні потреби та прагнення об'єктів навчання. Крім того, інтеграція технологічних інструментів може відігравати роль у персоналізації досвіду.

Сучасні технології штучного інтелекту (ШІ) кардинально змінюють підходи до освіти, дозволяючи створювати адаптивні та персоналізовані навчальні системи [2]. Автоматизовані та інтелектуальні навчальні системи сприяють підвищенню ефективності навчального процесу за рахунок врахування індивідуальних потреб студентів, автоматичного контролю знань та оптимізації навчальних програм. Одним із перспективних підходів є використання концепції задоволення обмежень (CSP) [3], що дозволяє ефективно розв'язувати завдання розкладання, персоналізованого навчання та автоматичної генерації тестів. Дослідження даної теми є актуальним у зв'язку з необхідністю підвищення гнучкості, адаптивності та інтелектуальності сучасних навчальних систем.

Сучасні інформаційні технології суттєво змінюють освітній процес, роблячи його більш гнучким, персоналізованим та ефективним. Використання штучного інтелекту (ШІ) у навчальних системах дозволяє покращити адаптацію освітнього контенту до потреб студентів, автоматизувати оцінювання знань, розробити персоналізовані маршрути навчання та забезпечити інтерактивний зворотний зв'язок.

Крім того, сучасні системи дистанційного навчання потребують ефективних механізмів автоматичної генерації тестів та контролю навчального процесу. Концепція CSP дає змогу розробляти алгоритми, що враховують складність питань, рівень підготовки студента та його попередні результати. Це сприяє підвищенню точності оцінювання знань та зниженню навантаження на викладачів.

Застосування ШІ в освітніх технологіях має велике значення не лише для академічної сфери, а й для корпоративного навчання, підготовки спеціалістів у різних галузях, розробки тренажерів для самонавчання. Впровадження інтелектуальних навчальних систем допоможе подолати

проблему масової освіти, забезпечивши кожному студенту унікальний освітній маршрут.

Таким чином, дослідження інтеграції концепції CSP у навчальні системи є актуальним у зв'язку з необхідністю підвищення ефективності освітнього процесу, оптимізації навчальних програм та автоматизації оцінювання знань. Результати роботи сприятимуть покращенню якості освіти шляхом впровадження адаптивного навчання та персоналізації контенту, що є одним із пріоритетних напрямів сучасної педагогіки та освітніх технологій.

Мета дослідження - розробка інтелектуальної навчальної системи на основі концепції задоволення обмежень (CSP) для підвищення ефективності процесу навчання та персоналізації освітнього контенту.

Об'єкт дослідження - автоматизовані та інтелектуальні навчальні системи, що використовують методи штучного інтелекту.

Предмет дослідження - методи та алгоритми концепції задоволення обмежень (CSP) у контексті побудови інтелектуальних навчальних систем.

Завдання дослідження

1. Проаналізувати існуючі автоматизовані та інтелектуальні навчальні системи, їхні особливості та підходи до персоналізації навчання.
2. Дослідити методи та алгоритми концепції задоволення обмежень (CSP) для навчальних систем.
3. Розробити модель інтелектуальної навчальної системи з використанням CSP.
4. Реалізувати прототип інтелектуальної навчальної системи з адаптивним навчальним процесом.

Методи дослідження

В роботі використано методи аналізу літературних джерел і існуючих навчальних систем, математичне моделювання та використання теорії обмежень у навчальному процесі, розробка алгоритмів на основі концепції CSP, програмна реалізація та експериментальне тестування запропонованої системи.

Апробація результатів магістерської роботи

Результати магістерського дослідження були опубліковані у фаховому науковому журналі Вісник Хмельницького національного університету (категорія “Б”, F2 - Інженерія програмного забезпечення (121)) в наступних статтях “Методологія оптимізації інтелектуальних систем на основі даних з правил впевненості” (<https://doi.org/10.31891/2307-5732-2025-351-10>) та “Розробка методу інтеграції машинного навчання з виведенням на основі правил” (<https://doi.org/10.31891/2307-5732-2025-351-12>)

Наукова новизна отриманих результатів

Запропоновано новий підхід до побудови інтелектуальних навчальних систем на основі концепції задоволення обмежень (CSP). Розроблено модель адаптивного персоналізованого навчання, що враховує когнітивні особливості студентів та динамічно змінює освітній контент. Реалізовано програмний прототип інтелектуальної навчальної системи з можливістю адаптації під індивідуальні потреби користувача.

Практичне значення результатів

Запропонована система може використовуватися у вищих навчальних закладах, онлайн-освіті та корпоративному навчанні для підвищення ефективності навчального процесу. Розроблений підхід може бути інтегрований у існуючі платформи дистанційного навчання для підвищення рівня автоматизації та адаптивності навчального середовища.

Структура магістерської роботи. Робота складається зі вступу, трьох розділів та висновків. Загальний обсяг роботи становить 93 сторінки, і містить 23 рисунки, список використаних джерел із 50 найменувань.

РОЗДІЛ 1. АНАЛІЗ КОНЦЕПЦІЙ ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ ПОБУДОВИ НАВЧАЛЬНИХ СИСТЕМ

1.1. Дослідження особливостей автоматизованих та інтелектуальних навчальних систем

Автоматизовані навчальні системи (АНС) – це програмні або апаратно-програмні комплекси, які призначені для підтримки та управління навчальним процесом із використанням інформаційних технологій. Вони спрямовані на автоматизацію викладання, контролю знань та адаптації навчального матеріалу відповідно до рівня підготовки студента [4 - 6].

АНС можуть функціонувати як самостійні платформи або бути частиною складніших інтелектуальних навчальних систем (ІНС), що використовують методи штучного інтелекту (ШІ)

Автоматизовані навчальні системи (АНС) мають ряд ключових особливостей, які визначають їхню ефективність та функціональність [11-13].

1. Адаптивність та персоналізація

- Система підлаштовується під рівень знань користувача.
- Використання ШІ для аналізу успішності та рекомендацій.
- Індивідуальні навчальні маршрути.

2. Інтерактивність та мультимедійність

- Використання анімацій, відео, аудіо, симуляцій.
- Практичні завдання, віртуальні лабораторії.
- Інтерактивні вправи (тести, квести, кейси).

3. Автоматичний контроль знань

- Генерація тестів та перевірка відповідей у реальному часі.
- Аналітика успішності студента (оцінки, прогрес).
- Виявлення слабких місць та рекомендації щодо навчання.

4. Доступність та мобільність

- Онлайн-доступ з будь-якого пристрою (ПК, планшет, смартфон).

- Можливість навчатися в будь-який час і в будь-якому місці.

- Хмарні технології для зберігання навчальних матеріалів.

5. Автоматизація навчального процесу

- Автоматичне розподілення матеріалів залежно від рівня студента.

- Планування розкладу занять і нагадування.

- Інтелектуальні чат-боти та віртуальні тьютори.

6. Гнучкість та масштабованість

- Легке оновлення та доповнення навчального контенту.

- Підтримка різних курсів і програм навчання.

- Можливість використання у школах, університетах та корпоративному навчанні.

7. Використання сучасних технологій

- Штучний інтелект (AI) для персоналізації.

- Великі дані (Big Data) для аналізу прогресу студентів.

- Віртуальна (VR) та доповнена реальність (AR) для імерсивного навчання.

Автоматизовані навчальні системи роблять освіту ефективнішою, доступнішою та індивідуальною, допомагаючи студентам навчатися у зручному темпі та форматі.

Переваги Автоматизованих навчальних систем

1. Гнучкість – доступ до навчальних матеріалів у будь-який час і з будь-якого місця.

2. Персоналізація – адаптація контенту відповідно до рівня знань студента.

3. Об'єктивність – автоматичне оцінювання без людського фактора.

4. Інтерактивність – використання мультимедіа, симуляторів, тестів.

5. Масштабованість – можливість навчання великої кількості студентів одночасно.

Автоматизовані навчальні системи значно покращують ефективність освітнього процесу, дозволяючи автоматизувати викладання, перевірку знань

та адаптацію контенту. Найбільш перспективним напрямом є інтелектуальні навчальні системи, що інтегрують штучний інтелект для персоналізованого навчання та глибокого аналізу поведінки студентів [8, 9].



Рис. 1.1. Класифікація автоматизованих навчальних систем

Автоматизовані навчальні системи (АНС) можна класифікувати за різними критеріями. Основні підходи до класифікації включають:

1. За рівнем адаптивності:

- Традиційні (жорсткі) АНС – використовують фіксовані навчальні матеріали та алгоритми навчання без урахування індивідуальних особливостей студентів.

- Адаптивні АНС – підлаштовуються під рівень знань, стиль навчання та інші характеристики студента.

- Інтелектуальні АНС – використовують методи штучного інтелекту, машинного навчання та видобування знань для персоналізації навчального процесу.

2. За способом взаємодії з користувачем:

- АНС для самостійного навчання – студенти взаємодіють із системою без безпосереднього контролю викладача.

- АНС для підтримки викладача – автоматизують окремі аспекти навчання, наприклад, тестування або надання додаткових матеріалів.

- Гібридні системи – поєднують елементи самостійного навчання та підтримки викладача.

3. За технологічним підходом:

- Лінійні системи – подають навчальний матеріал у фіксованій послідовності.

- Модульні системи – організовані за принципом незалежних модулів, які можуть комбінуватися залежно від потреб студента.

- Мережеві системи – функціонують у хмарному або локальному мережевому середовищі, дозволяючи одночасний доступ великої кількості користувачів.

4. За рівнем інтерактивності:

- Пасивні системи – студенти споживають навчальний контент без активної взаємодії.

- Інтерактивні системи – передбачають активну взаємодію, наприклад, у формі тестів, симуляцій або кейс-методів.

- Віртуальні навчальні середовища (VLE) – використовують віртуальну або доповнену реальність для створення реалістичного навчального досвіду.

5. За освітнім призначенням:

- Системи загальної освіти – використовуються у школах, університетах та інших навчальних закладах.

- Професійні навчальні системи – застосовуються для підготовки та підвищення кваліфікації спеціалістів у конкретних галузях.

- Тренажери та симулятори – призначені для відпрацювання практичних навичок (наприклад, для медицини, авіації чи військової справи).

б. За характером навчального контенту:

- Текстові АНС – використовують текстові матеріали як основний засіб навчання.

- Мультимедійні АНС – інтегрують текст, зображення, відео та аудіо для кращого засвоєння знань.

- Ігрові навчальні системи (edutainment) – застосовують ігрові механіки для підвищення мотивації до навчання.

Класифікація АНС може поєднувати кілька критеріїв одночасно, що дозволяє створювати комплексні навчальні середовища, адаптовані до різних потреб користувачів.

Інтеграція передових технологій у сектор освіти, таких як штучний інтелект (ШІ) та інтелектуальні освітні технології (ІОТ), значно розширила можливості для розробки індивідуальних програм. Ці інноваційні інструменти дозволяють вчителям створювати навчальні плани та методи викладання. Освітняни використовують ці ресурси для прогнозування успішності учнів, покращення процесу навчання та прийняття рішень на основі даних. Впроваджуючи технології в навчальне середовище, з'являється можливість сприяти участі учнів та розвивати критичне мислення та аналітичні здібності. Застосування ШІ в освіті охоплює різні напрямки, включаючи ресурси машинного навчання, віртуальних асистентів, чат-ботів, освітні ігри, симуляції віртуальної реальності та інтелектуальні освітні інструменти.

Інтелектуальні технології - це програми та системи, розроблені на основі методів штучного інтелекту. Основними характеристиками інтелектуальних технологій є адаптивність (здатність змінювати свою поведінку залежно від вхідних даних), автоматизація (виконання складних завдань без необхідності втручання людини), аналітика (поглиблений аналіз даних з метою виявлення закономірностей та прогнозування). Інтелектуальні освітні технології є частиною інтелектуальних технологій, спрямованих на покращення процесів навчання та викладання [15, 16].

Ці системи поєднують інструменти штучного інтелекту з освітньою аналітикою для створення персоналізованих, адаптивних та ефективних освітніх середовищ. Ключові аспекти таких технологій включають персоналізацію (налаштування контенту та підходів до навчання відповідно до потреб та стилю навчання кожного учня), моніторинг прогресу (використання аналітики даних для відстеження результатів та виявлення проблемних зон), адаптивне навчання (динамічне коригування навчальних матеріалів у режимі реального часу).

Інноваційні освітні технології трансформують освітній ландшафт за допомогою ШІ та аналізу даних. Ці інструменти проявляються у форматах, таких як програмне забезпечення, онлайн-платформи та мобільні додатки, спрямовані на надання учням та викладачам можливостей для залучення та отримання знань.

1.3. Концепції штучного інтелекту, що застосовуються для навчальних систем

Інтелектуальні навчальні системи (ІНС) використовують різні концепції штучного інтелекту (ШІ) для підвищення ефективності навчального процесу, персоналізації освітнього контенту та автоматизації оцінювання знань [5, 6]. Ось основні концепції ШІ, що застосовуються для навчальних систем (рис. 1.2).

1. Експертні системи в навчанні

Суть концепції: Експертні системи імітують процес мислення фахівця у певній предметній галузі та надають студенту рекомендації чи пояснення.

Застосування в навчальних системах:

- Автоматичний вибір навчального матеріалу залежно від знань студента.

- Надання роз'яснень та порад у режимі діалогу.

- Використання бази знань та механізму виведення для аналізу відповідей.



Рис. 1.2. Структуризація концепцій ШІ для використання в навчальних системах

2. Машинне навчання та глибоке навчання

Суть концепції: Використання алгоритмів машинного навчання (ML) та глибокого навчання (DL) для аналізу поведінки студентів і адаптації навчального процесу.

Застосування в навчальних системах:

- Персоналізоване навчання на основі історії навчання студента.
- Прогнозування результатів навчання та рекомендації щодо покращення.
- Автоматичне оцінювання тестів і письмових завдань за допомогою нейромереж.

3. Обчислювальний інтелект та еволюційні алгоритми

Суть концепції: Використання генетичних алгоритмів та нейроеволюції для оптимізації навчального процесу.

Застосування в навчальних системах:

- Автоматизоване формування навчальних маршрутів на основі попереднього досвіду студентів.

- Оптимізація тестових завдань для адаптивного тестування.

- Підбір ефективних стратегій викладання.

4. *Концепція представлення та задоволення обмежень (CSP - Constraint Satisfaction Problem) [7]*

Суть концепції: Використання методів CSP для формалізації навчальних задач у вигляді набору змінних, доменів значень та обмежень між ними.

Застосування в навчальних системах:

- Автоматизоване складання розкладу занять та іспитів.

- Формування персоналізованих навчальних траєкторій.

- Автоматична генерація тестових завдань з урахуванням складності та рівня підготовки студента.

5. Обробка природної мови (NLP - Natural Language Processing)

Суть концепції: Використання алгоритмів NLP для аналізу та генерації тексту, що допомагає покращити взаємодію студента із системою.

Застосування в навчальних системах:

- Автоматичне оцінювання есе та відкритих відповідей.

- Створення чат-ботів-тьюторів для супроводу навчального процесу.

- Розпізнавання голосових команд та генерація пояснень у текстовому форматі.

6. Інтелектуальні агенти та рекомендаційні системи

Суть концепції: Використання автономних агентів, що аналізують поведінку студента і рекомендують оптимальний навчальний матеріал.

Застосування в навчальних системах:

- Динамічний підбір навчального контенту відповідно до рівня знань студента.

- Індивідуалізовані рекомендації щодо наступних тем для вивчення.

- Виявлення прогалин у знаннях студента.

7. Комп'ютерний зір (Computer Vision) у навчальних системах

Суть концепції: Використання алгоритмів комп'ютерного зору для аналізу поведінки студентів під час навчання.

Застосування в навчальних системах:

- Автоматичний моніторинг уваги студентів під час онлайн-занять.

- Біометрична ідентифікація студентів під час тестування.

- Аналіз міміки та жестів для визначення рівня розуміння матеріалу.

Отже, штучний інтелект значно розширює можливості навчальних систем, роблячи їх адаптивними, персоналізованими та ефективними. Використання експертних систем, машинного навчання, CSP, NLP та інших підходів дозволяє автоматизувати процес навчання, покращити взаємодію студентів із системами та підвищити якість освіти.

1.3. Формування стратегії засвоєння знань у навчальних системах на основі штучного інтелекту

Оскільки інтелектуальні навчальні системи використовують методи ШІ, особливо експертних систем, то важливим питанням є дослідженням логічних зв'язків між елементами та різними концепціями знань *KnowledgesConceptions*, що є основою ефективного функціонування механізму інференції [11 - 13].

Одним із ключових аспектів реалізації автоматизованої навчальної системи, що використовує концепції штучного інтелекту, є вибір ефективної стратегії засвоєння знань суб'єктами навчання. В основі цього процесу лежить наявність у суб'єкта певної базової множини знань із відповідної предметної області, а також здатність розуміти механізми інференції, що

застосовуються системою. Це забезпечує можливість логічного переходу від уже засвоєного знання до нових концепцій. Такий підхід може бути структурований у межах наступних етапів:

- Перевірка засвоєного матеріалу на основі контролю релевантних обмежень, що накладаються на розв'язання навчальних проблем.
- Оновлення зв'язків між раніше засвоєними знаннями та новими елементами знань, що забезпечує їхню когнітивну інтеграцію.
- Аналіз семантичних взаємозв'язків нової інформації з уже наявними знаннями, що сприяє глибшому розумінню предметної області.
- Узагальнення раніше засвоєного матеріалу та нових знань у єдину систему концептуалізованих знань, що є основою для подальшого розширення когнітивного простору суб'єкта навчання.

1.3.1. Біхевіористична теорія навчання у контексті автоматизованих навчальних систем

Автоматизовані навчальні системи адаптують класичні навчальні теорії, зокрема біхевіористичний підхід, для застосування у комп'ютерно-орієнтованому освітньому середовищі. Основна концепція такої адаптації полягає в перенесенні принципів традиційного навчального процесу, що здійснюється за участі викладача, на автоматизовані системи, які використовують алгоритми штучного інтелекту для організації навчання [27].

Ключові аспекти адаптації біхевіористичної теорії в АНС включають:

- Ідентифікацію навчальних ситуацій та моделей поведінки суб'єкта навчання в межах конкретних навчальних завдань. Це передбачає реалізацію навчальних тренінгів, що містять типові приклади розв'язання задач і проблем, а також демонструють стандартизовані рішення.
- Формування поведінкових реакцій на навчальні стимули, що досягається через багаторазове повторення навчального матеріалу та закріплення шаблонних відповідей.

- Пряма залежність між інтенсивністю навчальних тренінгів та рівнем засвоєння знань: чим більша тривалість та складність навчального процесу, тим вищий рівень сформованих знань, умінь і навичок у суб'єкта навчання в системі.

Таким чином, автоматизовані навчальні системи, що базуються на принципах біхевіоризму, ефективно моделюють навчальний процес, оптимізуючи механізми засвоєння знань та адаптуючи їх до індивідуальних особливостей студентів.

1.3.2. Використання навчальних ефектів в інтелектуальних АНС на основі концепції представлення та задоволення обмежень

Функціонування інтелектуальних автоматизованих навчальних систем (АНС), що базуються на концепції представлення та задоволення обмежень, може бути формалізовано через аналог формули Скіннера [3].

$$\text{LearningSituation} \rightarrow \text{TraineeLearningActivities} \rightarrow \text{SystemConfirmation} [\text{ConstraintsSatisfaction}]$$

Ця формула описує взаємозв'язки між множиною навчальних ситуацій та навчальною діяльністю суб'єкта навчання в межах визначених ситуацій. Процес контролюється системою обмежень, які накладаються як на окремі навчальні ситуації, так і на їхню сукупність. Оцінювання успішності виконання навчальних завдань здійснюється за наступним принципом:

- якщо всі обмеження задоволені, система фіксує навчальну діяльність як успішну;
- якщо задоволена лише частина обмежень, результат вважається частково успішним;
- якщо кількість задоволених обмежень є нижчою за встановлений поріг, діяльність визначається як неуспішна.

Система підтверджує результати навчання шляхом аналізу задоволених і порушених обмежень, що дозволяє здійснювати глибшу оцінку процесу засвоєння знань [17, 18].

Відповідно до методики Скіннера, навчальний контент розподіляється на дискретні порції, що відображають окремі навчальні ситуації, структуру яких можна об'єктивно оцінити щодо ефективності навчального процесу. Зміст навчальних ситуацій формується як послідовність навчальних задач і проблем. Завдяки тому, що правильні відповіді та коректні рішення вже закладені в систему, вона може їх візуалізувати для суб'єкта навчання, забезпечуючи багаторазове повторення матеріалу, що сприяє закріпленню знань, умінь і навичок.

Цей підхід можна розглядати як цифрову адаптацію програмованих навчальних посібників Скіннера, які в класичній педагогічній теорії використовувалися для інтенсивного та ефективного навчання за короткий проміжок часу. Однак впровадження цієї методики в автоматизовані навчальні системи супроводжується певними проблемами:

- Механічність та одноманітність тестових завдань, що може знижувати мотивацію навчання.

- Відсутність логічної послідовності та системності викладу матеріалу через поділ контенту на численні дискретні фрагменти, що ускладнює формування цілісного уявлення про предмет.

- Необхідність постійної диверсифікації обмежень для збереження релевантності оцінювання рівня знань, особливо при повторному розв'язанні схожих завдань у різних контекстах.

- Обмежена адаптивність контенту, зумовлена значною кількістю повторень стандартних відповідей і рішень, що може знижувати ефективність навчального процесу.

Таким чином, хоча методика Скіннера демонструє високу ефективність у швидкому засвоєнні знань, її інтеграція в інтелектуальні АНС потребує подальших удосконалень, спрямованих на підвищення адаптивності

навчального процесу, різноманітності навчальних завдань та збереження логічної цілісності навчального контенту.

1.3.3. Адаптація підходу Краудера в автоматизованих навчальних системах

Адаптація методики Краудера [4] у контексті автоматизованих навчальних систем (АНС) передбачає класифікацію відповідей і рішень суб'єкта навчання за наступною схемою:

- Правильна відповідь (✓) – коректне рішення (✓)
- Неточна відповідь (△) – частково коректне рішення (△)
- Неправильна відповідь (✗) – некоректне рішення (✗)

Також рішень об'єкта навчання можна представити за наступною схемою:

- правильна відповідь ($CF = 100\%$) [коректне рішення ($CF = 100\%$)];
- неточна відповідь ($30\% < CF < 100\%$) [частково коректне рішення ($30\% < CF < 100\%$)];
- неправильна відповідь ($CF < 30\%$) [некоректне рішення ($CF < 30\%$)].

Ефективне функціонування АНС на основі цього підходу вимагає високого рівня адаптивності, що реалізується через трирівневу систему реагування:

- Перший рівень адаптації – активація коригувального та уточнювального контенту у разі неточної відповіді або частково коректного рішення. Це дозволяє суб'єкту навчання отримати додаткове пояснення, що сприяє формуванню більш точного розуміння матеріалу.

- Другий рівень адаптації – активація відповідних розділів основного довідкового матеріалу, контекстно-залежної допомоги або релевантних розділів бази даних навчального контенту у випадку неправильної відповіді або некоректного рішення. Цей рівень забезпечує доступ до поглибленого пояснення, спрямованого на корекцію помилок.

- Третій рівень адаптації – активація повідомлень про підтвердження правильності відповіді або коректності рішення, демонстрація оптимальних та альтернативних рішень навчальної проблеми, верифікація профілю суб'єкта навчання, визначення його зони найближчого розвитку та коригування навчальної стратегії відповідно до отриманих результатів [5, 6].

Таким чином, адаптація підходу Краудера в АНС дозволяє створювати динамічні навчальні середовища, що забезпечують персоналізоване навчання, підвищуючи його ефективність через диференційовану підтримку та гнучку корекцію навчального процесу.

1.4. Модель контролю навчального процесу

Основним завданням навчального процесу, що підтримується адаптивним навчальним середовищем (АНС), є забезпечення ефективної передачі знань. Успішність даного процесу залежить від наявності носія контентних знань, яким виступає експерт предметної області, викладач навчального курсу, а також від визначених способів представлення, трансформації та передачі знань. Важливим аспектом є також забезпечення ефективної взаємодії між об'єктами навчання, які виступають у ролі одержувачів знань [11, 12].

Роль викладача в АНС, особливо в його інтелектуальних версіях, є аналогічною до ролі викладача в класичному навчальному процесі. Вона полягає у створенні та підтримці належних умов навчання, що сприяють формуванню знань у студентів. Це включає подання навчального контенту у структурованому вигляді, встановлення асоціативних зв'язків між складовими навчального матеріалу та виокремлення ключових елементів контенту.

Контроль навчального процесу в АНС зазвичай здійснюється шляхом співставлення трьох ключових компонентів: поточного рівня знань об'єкта

навчання, моделі предметної області та його індивідуального навчального профілю в системі.



Рис. 1.3. Трьохкомпонентна модель контролю навчального процесу в АНС

Процес контролю здійснюється з урахуванням послідовності навчального процесу та результатів тестування знань за окремими темами навчального курсу, що відповідають певному маршруту, побудованому у вигляді послідовності вершин навчальної мережі. Система моніторингу навчальних досягнень фіксує результати тестування як за окремими темами та модулями, так і за всім навчальним курсом загалом. Крім того, система аналізує сумарні значення параметрів, що є частиною навчального профілю студента, реєструючи такі аспекти його діяльності, як вибір навчальної стратегії, послідовність розв'язуваних завдань, кількість звернень до навчального матеріалу, тривалість вивчення окремих тем, середній час роботи з одиницями контенту, а також часові показники виконання тестових завдань і навчальних проблем.

Особливістю навчального процесу в інтелектуальних АНС є використання методів видобування даних *DataMining* та знань *KnowledgeMining*. Дані технології дозволяють одержувати високий рівень інтелектуальної адаптивності АНС.

1.5. Опис мультимодальної моделі онлайн навчання

Згідно з [10], багато теоретиків розглядають онлайн-навчання як підкатегорію ‘загального навчання’ та ‘дистанційного навчання’ – перспектива, яка ускладнює побудову стандартної теорії онлайн-освіти. Тим не менш, в [10] обговорюється можливість розробки інклюзивної теорії онлайн-навчання на основі моделі СоІ (Спільнота дослідження). Він спирається на роботу, яка визначає ефективні навчальні середовища, що характеризуються перетином чотирьох взаємопов'язаних лінз: орієнтованість на знання, орієнтованість на учня, орієнтованість на спільноту та орієнтованість на оцінювання. Враховуючи це та той факт, що Інтернет еволюціонував і підтримує багато різних типів медіа, автор [10] пропонує нову модель онлайн-навчання, зображену на рисунку 1.4. У моделі ми бачимо акторів (студентів і викладачів), контент і взаємодію між ними. Студенти можуть отримати доступ до контенту або співпрацювати з викладачами для отримання допомоги через синхронні або асинхронні заходи та структуровані навчальні інструменти.

Варіант підходу [10], відомий як ‘Мультимодальна модель онлайн-освіти’, запропонована в [14], є нещодавньою спробою інтеграції онлайн-навчання. Ця парадигма запропонувала більш практичну основу для онлайн-навчання. На рисунку 1.4 зображена модель онлайн-навчання, яка демонструє ключові елементи та взаємодії в онлайн-освіті.

Центральні елементи:

- Студент: Знаходиться вгорі, представляючи головного учасника навчального процесу.

навчальних посібників, симуляцій, ігор, віртуальних лабораторій та електронних книг.

- Студент-Викладач: Демонструє взаємодію між студентом та викладачем, що може відбуватися асинхронно (наприклад, електронна пошта, форуми) або синхронно (наприклад, відеоконференції, чати).

- Студент-Студент: Відображає співпрацю та взаємодію між студентами, включаючи спільне навчання та обговорення.

- Викладач-Контент: Показує, як викладач використовує та взаємодіє з навчальним матеріалом для створення та організації навчального процесу.

- Викладач-Викладач: Демонструє взаємодію між викладачами, включаючи обмін досвідом та співпрацю.

- Інші Викладачі: Вказує на ширшу мережу викладачів, які можуть надавати підтримку та ресурси.

Додаткові аспекти:

- Незалежне навчання: Показано праворуч, підкреслюючи важливість самостійного навчання студента.

- Спільнота дослідження: Показано зліва, підкреслюючи важливість створення спільноти для спільного навчання та обговорення.

- Темп, спільне навчання: Показано зліва, вказуючи на структуроване та спільне навчання.

- Підтримка однолітків, сім'ї та професійна підтримка: Показано праворуч, вказуючи на важливість підтримки з різних джерел.

- Структуровані навчальні ресурси: Показано праворуч, підкреслюючи важливість організації та надання якісних навчальних матеріалів.

- Комунікація: Показано зліва, підкреслюючи важливість ефективної комунікації в онлайн-навчанні.

Загалом, дана модель показує складну мережу взаємодій та елементів, які складають ефективне онлайн-навчання. Вона підкреслює важливість взаємодії між студентами, викладачами та контентом, а також важливість підтримки, спільноти та структурованих ресурсів.

Онтологічні припущення моделі стверджують, що знання існують розподілено в мережі акторів, контенту та взаємодій/інструментів. Вона передбачає об'єктивний погляд на знання, до яких можна отримати доступ і якими можна ділитися через онлайн-мережу. Що стосується епістемологічних припущень, то модель передбачає, що знання конструюються через активну взаємодію та участь у мережі, і приймає прагматичний, прикладний погляд, що знання демонструються та застосовуються на практиці. Модель узгоджується з соціально-конструктивістськими теоріями, такими як CoI, які розглядають навчання як соціально сконструйоване через взаємодію та дискурс. Крім того, вона сумісна з теоріями, що наголошують на змішаних/мультиmodalних навчальних середовищах, таких як Мультиmodalна модель.

Висновки до розділу

У першому розділі проведено аналіз основних концепцій штучного інтелекту, які застосовуються для побудови навчальних систем. Зокрема, досліджено особливості автоматизованих та інтелектуальних навчальних систем (ІНС), що дозволяють значно підвищити ефективність освітнього процесу шляхом персоналізації навчального матеріалу та використання адаптивних алгоритмів. Розглянуто різні стратегії засвоєння знань, що ґрунтуються на штучному інтелекті. Особливу увагу приділено біхевіористичній теорії навчання, яка застосовується для побудови автоматизованих навчальних систем, та використанню навчальних ефектів в інтелектуальних системах на основі концепції представлення та задоволення обмежень (CSP). Доведено, що такий підхід забезпечує гнучке адаптування навчального процесу до індивідуальних особливостей користувача.

Також проаналізовано адаптацію підходу Краудера в автоматизованих навчальних системах, що дозволяє покращити інтерактивність навчання та підвищити рівень залученості користувачів.

Окрему увагу приділено питанням контролю навчального процесу, що є важливим елементом будь-якої навчальної системи. Запропонована модель контролю знань базується на механізмах збору даних про дії користувачів та аналізу їхнього прогресу за допомогою інтелектуальних алгоритмів.

Нарешті, у розділі розглянуто мультимодальну модель онлайн-навчання, що передбачає інтеграцію різних форматів навчальних матеріалів (текст, відео, інтерактивні елементи), що сприяє підвищенню ефективності засвоєння знань.

Таким чином, проведене дослідження підтвердило доцільність використання штучного інтелекту для побудови навчальних систем, що дозволяє забезпечити персоналізацію навчального процесу, підвищити його ефективність та автоматизувати контроль знань

РОЗДІЛ 2. ДОСЛІДЖЕННЯ МЕТОДІВ ТА АЛГОРИТМІВ КОНЦЕПЦІЇ ЗАДОВОЛЕННЯ ОБМЕЖЕНЬ (CSP) В КОНТЕКСТІ ПОБУДОВИ НАВЧАЛЬНИХ СИСТЕМ

2.1. Особливості концепції задоволення обмежень (CSP)

Constraint Satisfaction Problem (CSP) – це клас задач у штучному інтелекті (ШІ) та дискретній математиці, де рішення шукається шляхом задоволення заданих обмежень (constraints). Ці задачі часто виникають у плануванні, розкладанні, розпізнаванні образів, конфігурації систем тощо [7].

Розглянемо формальне визначення CSP. Задача CSP визначається як триплет:

$$CSP = (X, D, C)$$

де:

X - множина змінних: $X = \{X_1, X_2, \dots, X_n\}$

D - множина доменів значень: кожна змінна X_i має кінцевий набір можливих значень D_i ,

C - множина обмежень (constraints): кожне обмеження визначає допустимі комбінації значень для підмножини змінних [18, 20].

Розглянемо завдання: знайти набір значень змінних із відповідних доменів, які задовольняють усі обмеження. Представимо приклад CSP на основі задачі розфарбовування карти (Map Coloring Problem):

- Змінні: $X = \{A, B, C, D\}$ - області карти.

- Домен: $D = \{\text{червоний, зелений, синій}\}$.

Обмеження: сусідні області повинні мати різні кольори, наприклад:

$$A \neq B, B \neq C, C \neq D, D \neq A.$$

Рішенням буде допустима розфарбовка, наприклад:

A = червоний, B = зелений, C = червоний, D = синій

Обмеження у CSP можна класифікувати за кількістю змінних:

- Унарні ($C(X_i)$) – обмеження на одну змінну (наприклад, $X_1 \neq \text{червоний}$).
- Бінарні ($C(X_i, X_j)$) – обмеження між двома змінними (наприклад, $X_1 \neq X_2$).
- N-арні ($C(X_1, X_2, \dots, X_n)$) – обмеження для більш ніж двох змінних (наприклад, у задачі латинського квадрата).

Розв'язування CSP може здійснюватися різними методами [21, 22]:

1) Пошукові методи

Backtracking Search (пошук з поверненням) – класичний рекурсивний алгоритм, що перебирає можливі значення змінних і повертається назад при порушенні обмежень.

Forward Checking (перевірка наперед) – розширення пошуку з поверненням, яке виключає неможливі значення ще до призначення змінних.

Constraint Propagation (поширення обмежень) – використання методів, таких як Arc Consistency (AC-3), щоб ще більше зменшити простір пошуку.

2) Локальні методи

Hill Climbing, Simulated Annealing – використовуються для пошуку прийнятних, хоча й не завжди оптимальних рішень.

Min-Conflicts Heuristic – евристичний алгоритм, який зменшує кількість порушень обмежень.

3) Алгебраїчні методи

Boolean Satisfiability Problem (SAT) – перетворення CSP у булеву форму та розв'язання SAT-розв'язувачами.

Integer Linear Programming (ILP) – формулювання CSP у вигляді лінійної програми з цілочисловими змінними.

Для прискорення пошуку рішення використовують (оптимізація розв'язання CSP):

Евристики вибору змінних:

- Most Constrained Variable (MRV) – спочатку вибираються змінні з найменшою кількістю допустимих значень.

- Degree Heuristic – обирається змінна, що найбільше пов'язана з іншими змінними.

Евристики вибору значень:

- Least Constraining Value (LCV) – вибирається значення, яке залишає найбільше можливостей для інших змінних.

Обробка обмежень перед пошуком:

- Arc Consistency (AC-3) – алгоритм, який прибирає значення, що порушують обмеження на рівні дуг (пар змінних).

- Path Consistency, k-Consistency – розширені методи забезпечення узгодженості.

CSP можна застосовувати в наступних реальних задачах:

- Розклад занять (Timetabling) – пошук розкладу, що уникає конфліктів.

- Задачі маршрутизації – наприклад, складання графіків поїздів чи транспорту.

- Генерація кросвордів – заповнення сітки слів із дотриманням буквосполучень.

- Автоматична перевірка конфігурацій – наприклад, конфігурація комп'ютерного обладнання.

- Розв'язання sudoku – класична CSP-задача з числовими обмеженнями.

2.2. Формалізоване представлення концепції задоволення обмежень в термінах навчальних систем

Проблема задоволення обмежень полягає в накладенні певних вимог на скінченну множину змінних в формі обмежень. Множина можливих значень (доменів) для кожної змінної є скінченною. Кожне обмеження вказує які кортежі значень є дозволеними для кожної підмножини базових змінних. Обмеження може бути задано в явній формі, через перелічення дозволених кортежів або не явно через алгебраїчні вирази [7, 13, 15].

Означення 2.1 (навчальна проблема на основі обмежень)

Навчальною проблемою на основі обмежень (*LCSP*) будемо вважати кортеж $LP = (V, D, C^S, C^V)$, де

$V = \{v_1, \dots, v_n\}$ - множина змінних, що називаються доменними змінними навчальної проблеми;

$D = \{D_1, \dots, D_n\}$ - є множиною доменів навчальної проблеми. Кожен домен є скінченною множиною, що містить можливі значення для відповідних змінних;

$C^S = \{c_1^s, \dots, c_n^s\}$ - множина обмежень релевантних до поточної навчальної проблеми, що задовольняються; кожне таке обмеження c_i^s є відношенням означеним на підмножині $\{v_{i_1}, \dots, v_{i_{k_i}}\}$ всіх змінних, таких що $\{D_{i_1} \times \dots \times D_{i_{k_i}}\} \supseteq c_i^s$;

$C^V = \{c_1^v, \dots, c_n^v\}$ - множина обмежень релевантних до поточної навчальної проблеми, що порушуються; кожне таке обмеження c_j^v є відношенням означеним на підмножині $\{v_{j_1}, \dots, v_{j_{l_j}}\}$ всіх змінних, таких що $\{D_{j_1} \times \dots \times D_{j_{l_j}}\} \supseteq c_j^v$

Проте на більш загальному рівні кожне з введених типів обмежень можна розглядати як обмеження релевантного типу $c^r = c^s \cup c^v$. Множину змінних в обмеженні c^r позначимо через V_{c^r} .

Якщо множина V_{c^r} містить один чи два елемента, то будемо їх називати унарними та бінарними обмеженнями навчальних проблем відповідно. Всі інші обмеження називатимемо не-бінарними. Навчальну проблему на основі обмежень будемо називати унарною (бінарною) якщо всі її обмеження є унарними (бінарними). Бінарні навчальні проблеми на основі обмежень відіграють особливу роль, оскільки кожен навчальну проблему можна перетворити у еквівалентну їй бінарну навчальну проблему, шляхом бінаризації обмежень. Проте на практиці, таке перетворення часто вимагає

надто багато додаткових змінних з великими доменами, що може складно ускладнити пошук рішення нових навчальних проблем. Структура навчальних проблем на основі обмежень може бути представлена графом обмежень.

Означення 2.2 Присвоєнням на множині навчальних проблем (пропонованим рішенням навчальної проблеми – *SubmittedSolution*) LP^{set} будемо вважати відображення Ψ^{LP} з множини змінних домену $V_I \subseteq V$ в їх відповідні домени, тобто $\Psi^{LP}(v_i) \in D_i$ для $v_i \in V_I$. Множину всіх присвоєнь для V_I будемо позначати $\Psi_{V_I}^{LP}$.

Будемо розглядати змінні в V_I в деякому довільному фіксованому порядку (v_1, \dots, v_k) . Тоді присвоєння Ψ^{LP} , що відповідає $\{(v_1, d_{v_1}), \dots, (v_k, d_{v_k})\}$ будемо записувати у вигляді $\Psi^{LP} = (d_{v_1}, \dots, d_{v_k})$ і будемо розглядати його як кортеж значень $(d_{v_1}, \dots, d_{v_k})$.

Присвоєння, означене на множині доменних змінних V будемо вважати повним, в іншому випадку – частковим. Множину всіх можливих повних присвоєнь $\Psi_V^{LP} \equiv D_1 \times \dots \times D_n$ – будемо називати простором рішень навчальної проблеми LP . Тобто всі можливі потенційні розвязки будуть знаходитись в цьому просторі.

Означення 2.3 Будемо вважати, що обмеження c^r задовольняється присвоєнням Ψ^{LP} (позначатимемо це через $\Psi^{LP} \models c^r$), якщо всі його змінні отримують такі значення, що відповідний кортеж значень належить c^r . В протилежному випадку обмеження c^r розглядається як незадоволене.

Для кожного заданого обмеження $c_i^r(v_{i_1}, \dots, v_{i_k})$ будемо позначати через $\neg c_i^r(v_{i_1}, \dots, v_{i_k})$ обмеження, яке незадовольняється коли c_i^r задовольняється.

Часткове присвоєння Ψ^{LP} будемо вважати послідовним, якщо всі обмеження, що відносяться тільки до змінних, що отримали присвоєння через Ψ^{LP} є задоволеними.

Означення 1.4 (рішення навчальної проблеми). Рішенням навчальної проблеми на основі обмежень (V, D, C^R) будемо вважати послідовне повне присвоєння, таке, що всі обмеження з множини C^R повинні бути задоволені.

Якщо навчальна проблема немає жодного рішення, то таку навчальну проблему будемо називати надобмеженою (переобмеженою) або непослідовною. Відповідно навчальну проблему яка має більше одного рішення будемо називати не достатньо-обмеженою. Обмеження є послабленим якщо в відношення можна додати додаткові елементи. Відповідно, якщо такі додані елементи видалити з відношення – то обмеження будемо вважати стиснутим.

Отже, CSP – це потужний підхід у штучному інтелекті, який використовується для моделювання та розв'язання комбінаторних задач. Завдяки ефективним алгоритмам та евристикам він застосовується у багатьох реальних областях, від логістики до розкладів та штучного інтелекту [18 -20].

2.3. Використання концепції CSP для побудови навчальних систем та систем тестування знань

Constraint Satisfaction Problem (CSP) можна застосовувати в навчальних системах та системах тестування знань для оптимізації процесів складання розкладу, генерації тестів, персоналізованого навчання тощо [23, 25].

Розглянемо використання CSP у навчальних системах. CSP можна застосувати для вирішення кількох ключових завдань у навчальному процесі:

2.3.1. Автоматизоване складання розкладу

Задача: Автоматизоване складання розкладу занять, яке враховує такі обмеження:

- Викладачі не можуть вести дві пари одночасно.
- Студенти не можуть мати перетину у розкладі.
- Аудиторії мають бути доступні.

- Спеціальні вимоги (наприклад, лабораторії для певних дисциплін).

CSP-модель:

- Змінні: заняття (лекції, семінари, лабораторні).
- Домен: набір можливих слотів часу та аудиторій.

Обмеження:

- Кожен викладач має власний графік і не може бути у двох місцях одночасно.
- Аудиторії не можуть бути зайняті більше ніж одним заняттям у певний час.
- Для практичних занять потрібні спеціальні аудиторії.

2.3.2. Задача персоналізованого навчання

Задача: Генерація персоналізованої траєкторії навчання відповідно до рівня знань студента.

CSP-модель:

- Змінні: теми, які потрібно вивчити.
- Домен: можливі навчальні матеріали, що підходять для рівня студента.

Обмеження:

- Студент повинен пройти теми у певній послідовності (наприклад, спочатку основи, потім складніші теми).
- Теми можуть залежати одна від одної (наприклад, 'Алгоритми' перед 'Структурами даних').
- Доступний час навчання студента.

2.3.3. Використання CSP у системах тестування знань

CSP може допомогти у створенні адаптивних тестів, які:

- Враховують рівень знань студента.
- Генерують унікальні тести для кожного учня.
- Забезпечують рівномірний розподіл складності питань.

Адаптивне тестування (CAT – Computerized Adaptive Testing)

Задача: Динамічне коригування складності тесту на основі відповідей студента.

CSP-модель:

- Змінні: рівень знань студента після кожного запитання.
- Домен: список наступних питань відповідно до рівня.

Обмеження:

- Якщо студент відповів правильно, наступне питання складніше.
- Якщо неправильно — питання стає простішим.
- Мінімальна кількість питань, щоб точно оцінити знання.

Методи реалізації:

- Використання моделі Item Response Theory (IRT) для підбору оптимальних питань.

- Використання CSP для динамічного вибору наступного питання.

Додаткові можливості CSP у навчальних системах

- Автоматичне групування студентів для спільної роботи за рівнем знань.

- Рекомендаційні системи для вибору навчальних матеріалів.

- Оптимізація ресурсів навчального закладу.

Отже, CSP дозволяє автоматизувати складні задачі в навчальних системах та системах тестування. Його використання забезпечує персоналізацію навчання, адаптивне тестування та оптимізацію ресурсів [29].

2.4. Застосування концепції CSP для задачі автоматичної генерації тестів

Розглянемо наступну задачу формування тесту, що відповідає визначеним критеріям складності та покриває всі необхідні теми [30, 31].

CSP-модель:

- Змінні: питання тесту.

- Домен: набір усіх доступних питань.

Обмеження:

- Тест має включати питання різної складності (легкі, середні, важкі).

- Питання мають покривати всі необхідні теми.

- Час на проходження тесту не повинен перевищувати певного ліміту.

Приклад алгоритму:

- Вибір питань відповідно до теми.

- Застосування обмежень щодо складності.

- Формування унікального варіанту тесту.

Наведемо приклад коду на Python для автоматичної генерації тесту за допомогою CSP. Ми використовуємо `python-constraint`, щоб сформувати тест із заданими обмеженнями [32].

Графічне представлення задачі автоматичної генерації тестів показано на рис. 2.1.

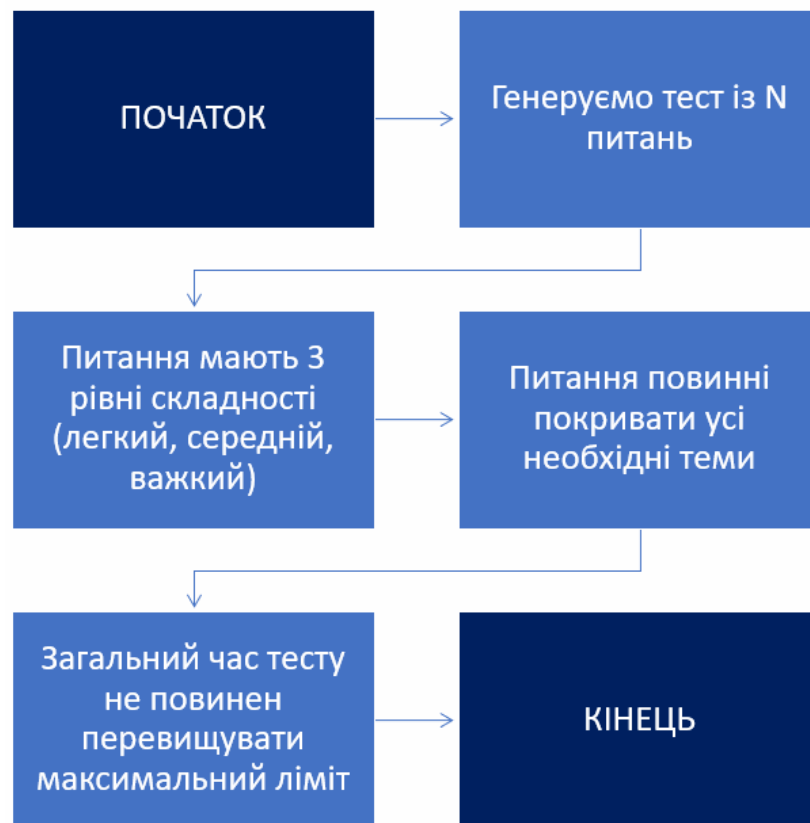


Рис. 2.1. Графічне представлення задачі автоматичної генерації тестів

Нижче в лістингу 2.1 представлена реалізація CSP у Python

Лістинг 2.1. реалізація CSP у Python

```
from constraint import Problem

# База питань: (ID, тема, складність, час на відповідь)
questions = [
    (1, 'Алгоритми', 'легкий', 2),
    (2, 'Алгоритми', 'середній', 3),
    (3, 'Алгоритми', 'важкий', 5),
    (4, 'Структури даних', 'легкий', 2),
    (5, 'Структури даних', 'середній', 3),
    (6, 'Структури даних', 'важкий', 5),
    (7, 'Бази даних', 'легкий', 2),
    (8, 'Бази даних', 'середній', 3),
    (9, 'Бази даних', 'важкий', 5)
]

# Налаштування тесту
N = 3 # Кількість питань у тесті
MAX_TIME = 10 # Максимальний час на проходження тесту
# Ініціалізація CSP-проблеми
problem = Problem()
# Додаємо змінні (питання тесту)
for i in range(N):
    problem.addVariable(f'Q{i}', questions)
# Обмеження 1: Всі питання повинні бути унікальними
def unique_questions(*args):
    return len(args) == len(set(args))
problem.addConstraint(unique_questions, [f'Q{i}' for i in
range(N)])
# Обмеження 2: Питання повинні покривати всі теми
def cover_all_topics(*args):
    topics = {q[1] for q in args}
    return len(topics) == 3 # 3 унікальні теми
problem.addConstraint(cover_all_topics, [f'Q{i}' for i in
range(N)])
```

```

# Обмеження 3: Загальний час тесту не повинен перевищувати
MAX_TIME

def within_time_limit(*args):
    total_time = sum(q[3] for q in args)
    return total_time <= MAX_TIME

problem.addConstraint(within_time_limit, [f'Q{i}' for i in
range(N)])

# Пошук розв'язку
solutions = problem.getSolutions()

# Виведення першого знайденого рішення (або всіх)
if solutions:
    print('Знайдено варіант тесту:')
    for i in range(N):
        print(f'Питання {i+1}: {solutions[0][f'Q{i}']}'')
else:
    print('Немає можливих варіантів тесту.')

```

Детально розглянемо як працює код показаний в лістингу 2.1.

1. Визначаємо набір питань із характеристиками:

- Тема
- Рівень складності
- Час на відповідь

2. Задаємо обмеження:

- Всі питання мають бути унікальними.
- Усі теми повинні бути покриті.
- Загальний час тесту не може перевищувати MAX_TIME.

3. Використовуємо CSP для генерації оптимального варіанту тесту.

4. Виводимо знайдене рішення (або повідомлення, якщо немає варіантів).

В лістингу 2.2 наведено приклад виводу

Лістинг 2.2. Приклад виводу

Знайдено варіант тесту:

Питання 1: (1, 'Алгоритми', 'легкий', 2)

Питання 2: (5, 'Структури даних', 'середній', 3)

Питання 3: (9, 'Бази даних', 'важкий', 5)

Цей тест покриває всі 3 теми, має різні рівні складності і не перевищує заданий час.

Можливі покращення представленого рішення є наступними:

- Адаптивне тестування → Динамічно змінювати питання на основі відповідей.
- Додавання інших параметрів → Наприклад, рівень студента, щоб підбирати питання точніше.
- Використання AI для генерації питань → Інтеграція з NLP для створення нових питань.

2.5. Програмна реалізація персоналізованого навчання на основі CSP

Нехай, потрібно сформувані персоналізований навчальний план для студента з урахуванням [34, 35]:

- Обмеженого часу на навчання.
- Послідовності тем (деякі теми є передумовами для інших).
- Рівня знань студента (легкий, середній, просунутий).

Лістинг 2.3. Реалізація CSP у Python

```
from constraint import Problem

# База навчальних тем: (ID, Назва, Складність, Час на
вивчення, Передумови)
topics = {
    1: ('Основи Python', 'легкий', 3, []),
    2: ('Функції в Python', 'середній', 4, [1]),
    3: ('Рекурсія', 'середній', 5, [2]),
    4: ('ООП в Python', 'просунутий', 6, [2]),
```

```

5: ('Алгоритми сортування', 'середній', 5, [1]),
6: ('Динамічне програмування', 'просунутий', 7, [3, 5]),
7: ('Бази даних SQL', 'легкий', 4, []),
8: ('Запити SQL', 'середній', 5, [7]),
9: ('Індекси та оптимізація SQL', 'просунутий', 6, [8])
}
# Параметри студента
STUDENT_LEVEL = 'середній' # легкий, середній, просунутий
AVAILABLE_TIME = 10 # Час на навчання
# Вибираємо теми, відповідні рівню студента
filtered_topics = {k: v for k, v in topics.items() if v[1]
in ['легкий', STUDENT_LEVEL, 'просунутий']}
# Ініціалізація CSP
problem = Problem()
# Додаємо змінні (обрані теми)
problem.addVariables('T', list(filtered_topics.keys()))
# Обмеження 1: Усі теми повинні мати виконані передумови
def prerequisites_satisfied(*chosen_topics):
    chosen_set = set(chosen_topics)
    for topic in chosen_topics:
        required = set(topics[topic][3]) # Передумови
        if not required.issubset(chosen_set): # Чи є всі
передумови?
            return False
    return True
problem.addConstraint(prerequisites_satisfied, ['T'] *
len(filtered_topics))
# Обмеження 2: Загальний час навчання не повинен
перевищувати AVAILABLE_TIME
def within_time_limit(*chosen_topics):
    total_time = sum(topics[t][2] for t in chosen_topics)
    return total_time <= AVAILABLE_TIME
problem.addConstraint(within_time_limit, ['T'] *
len(filtered_topics))
# Пошук розв'язку
solutions = problem.getSolutions()

```

```

# Вивід першого знайденого рішення
if solutions:
    best_plan = solutions[0]
    print('Персоналізований навчальний план:')
    for topic in sorted(best_plan.values()): # Вивід тем у
логічному порядку
        print(f'- {topics[topic][0]} ({topics[topic][2]}
годин)')
else:
    print('Неможливо скласти навчальний план з даними
обмеженнями.')

```



Рис. 2.2. Алгоритмічна схема формування персоналізованого навчального плану

В лістингу 2.4 наведено приклад виводу (для середнього рівня студента, 10 годин навчання).

Лістинг 2.4. Виведення плану навчання

Персоналізований навчальний план:

- Основи Python (3 годин)
- Функції в Python (4 годин)
- Алгоритми сортування (5 годин)

Теми підбрані так, щоб виконувалися передумови (немає тем без необхідної бази) і загальний час не перевищував доступний.

Також можливі покращення даного алгоритму:

- Додати вагові коефіцієнти (наприклад, 'бажані теми' отримують вищий пріоритет).
- Інтегрувати з AI для персоналізації на основі минулих тестів студента.
- Генерувати адаптивний план у реальному часі на основі відповідей під час навчання [36].

Після формування бази правил процедура логічного виведення застосовується для генерації вихідних результатів системи. Процес логічного виведення є багатостадійним і включає послідовні етапи, зокрема трансформацію вхідних даних, активацію правил, обчислення ваг активації, оновлення ступенів упевненості та агрегацію правил із використанням підходу доказового міркування (рис. 2.3).

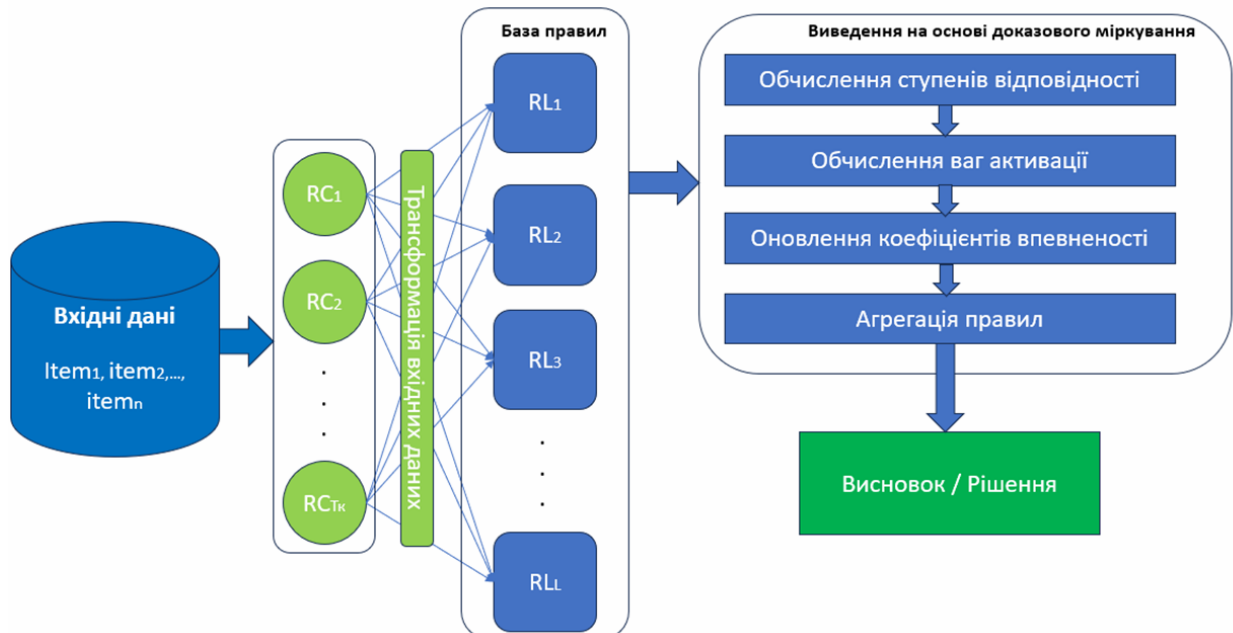


Рис. 2.3. Робочий процес побудови рішення в системі [50]

На етапі трансформації входу вхідні змінні проєктуються на референтні значення антецедентних атрибутів, у результаті чого визначається ступінь їх відповідності. Далі правила впевненості групуються у вигляді пакетів

антецедентів, а ваги їх активації обчислюються на основі отриманих ступенів відповідності. За диз'юнктивного припущення всі рівні відповідності агрегуються шляхом підсумовування, що дозволяє коректно відобразити семантику логічного оператора «АБО» у правилах впевненості.

Слід зауважити, що обчислення повного значення ваги активації правила з використанням операцій множення, додавання та ділення є обчислювально складним завданням. Це зумовлено обмеженнями традиційних обчислювальних підходів. Натомість застосування асоціативної пам'яті забезпечує можливість ефективної реалізації таких операцій завдяки її здатності до збереження, зіставлення та паралельної обробки патернів.

Висновки до розділу

У другому розділі досліджено методи та алгоритми концепції задоволення обмежень (CSP) в контексті побудови навчальних систем. Проведений аналіз показав, що CSP є ефективним підходом для моделювання та вирішення низки задач у сфері освіти, що вимагають автоматизації та адаптації до потреб користувача.

Розглянуто особливості концепції задоволення обмежень, її формалізоване представлення у термінах навчальних систем, що дозволяє структурувати знання, встановлювати взаємозв'язки між навчальними об'єктами та забезпечувати коректність навчального процесу.

Особливу увагу приділено використанню CSP у побудові навчальних систем та систем тестування знань. Доведено, що цей підхід ефективний у таких задачах, як:

- автоматизоване складання розкладу, що дозволяє оптимізувати розподіл навчальних занять відповідно до ресурсних обмежень.
- персоналізоване навчання, що передбачає адаптацію навчального матеріалу до індивідуальних особливостей студентів.

- тестування знань, де CSP використовується для визначення оптимальної послідовності завдань відповідно до компетентності студентів.

Окремо розглянуто можливість застосування концепції CSP для автоматичної генерації тестів, що дозволяє підвищити об'єктивність оцінювання знань, а також забезпечити унікальність і релевантність тестових завдань. Також у розділі представлено програмну реалізацію персоналізованого навчання на основі CSP, що демонструє практичну ефективність цього підходу у створенні адаптивних навчальних систем.

Таким чином, дослідження показало, що концепція задоволення обмежень є потужним інструментом для розробки інтелектуальних навчальних систем, спрямованих на автоматизацію навчального процесу, персоналізацію навчання та ефективне тестування знань.

РОЗДІЛ 3. ІМПЛЕМЕНТАЦІЯ КОНЦЕПЦІЙ ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ ПОБУДОВИ НАВЧАЛЬНОЇ СИСТЕМИ

3.1. Реалізація структурної схеми навчальної системи

Навчальна система на основі концепцій штучного інтелекту (в даному випадку на основі CSP) є одним із різновидів автоматизованих навчальних систем, що забезпечує реалізацію спрямованого та персоналізованого навчального процесу. Її функціонування ґрунтується на механізмах зворотного зв'язку із об'єктами навчання, використанні методів експертних систем та застосуванні інтелектуальних алгоритмів у процесі вирішення навчальних завдань [37, 39].



Рис. 3.1. Структурна схема та компоненти навчальної система на основі концепцій ШІ

Аналіз існуючих інтелектуальних систем показує, що навчальні системи цього класу зазвичай складається з чотирьох основних модулів:

1. Інтерфейсний модуль
2. Експертний модуль
3. Модуль об'єкта навчання

4. Навчальний модуль

Інтерфейсний модуль відіграє ключову роль у забезпеченні доступу користувачів до ресурсів системи та підтримує взаємодію як окремих студентів, так і групових навчальних процесів. Виділяють два основних підходи до побудови інтерфейсу:

- Класичний графічний інтерфейс (GUI) – найбільш поширений варіант, що використовується в традиційних та інтелектуальних АНС. Він забезпечує зручний доступ до навчального контенту, тестових завдань та інших навчальних матеріалів, а також підтримує інтерактивну взаємодію між студентом і системою .

- Симуляційне середовище – дозволяє створювати віртуальні моделі динамічних об'єктів і процесів предметної області. Наприклад, для навчання в нафтогазовій промисловості можуть застосовуватися симулятори технологічних процесів, що дозволяють студентам відпрацьовувати дії у змодельованих виробничих умовах.

Експертний модуль реалізує підтримку предметної області за допомогою експертних і штучно-інтелектуальних методів. Він ґрунтується на формалізації знань та експертного досвіду, зазвичай у вигляді системи правил або обмежень, які утворюють онтологічну модель предметної області. Основні компоненти цього модуля включають:

- Формалізовану систему об'єктів, суб'єктів, правил і зв'язків між ними.

- Систему обмежень, що визначають можливі взаємодії у межах предметної області.

- Засоби діагностики та коригування знань у разі невідповідності чи помилок у навчальному процесі.

Одним із прикладів застосування експертного модуля є автоматизована діагностика прогресу навчання студента або стану технічних систем, яка може виконувати аналіз нештатних ситуацій та пропонувати відповідні коригувальні дії.

Модуль об'єкта навчання забезпечує ведення та оновлення індивідуального навчального профілю студента. Він містить такі основні параметри:

- Поточний рівень засвоєння навчального матеріалу.
- Підсумковий рівень знань, умінь та навичок.
- Описи та еталонні зразки ідеальної експертної поведінки у межах предметної області.

На основі цих параметрів система здійснює постійний моніторинг навчального прогресу студента, формує персоналізовані рекомендації та застосовує механізми зворотного зв'язку для усунення прогалин у знаннях.

Функціональність навчального модуля полягає у формуванні та реалізації коригувальних впливів у рамках механізму зворотного зв'язку. З погляду студента ці впливи проявляються у вигляді навчальних інструкцій, адаптованих під його індивідуальні потреби.

Основні складові навчального модуля:

- Ідеальні моделі експертної поведінки, що містять стандартизовані зразки оперування з об'єктами предметної області.
- Алгоритми аналізу відхилень, які визначають розбіжності між діями студента та еталонною поведінкою експерта.
- Механізми адаптації навчального процесу, які забезпечують коригування завдань та навчальних матеріалів залежно від рівня знань студента.
- Ефективність роботи навчального модуля значною мірою залежить від точності порівняння індивідуального навчального профілю студента із зразками експертної діяльності, а також від правильності визначення та усунення прогалин у знаннях.

Загальна ефективність інтелектуальної навчальної системи значною мірою визначається якістю реалізації її структурних компонентів. Взаємодія між інтерфейсним, експертним, навчальним модулями та модулем об'єкта навчання дозволяє створювати адаптивні навчальні середовища, що

враховують особливості кожного студента, забезпечують індивідуалізацію навчального процесу та підвищують ефективність засвоєння знань [45].

Подальший розвиток інтелектуальних навчальних систем пов'язаний із удосконаленням методів штучного інтелекту, використанням технологій аналізу великих даних та застосуванням когнітивних моделей для побудови ще більш гнучких та персоналізованих навчальних систем.

3.2. Представлення моделей модулів інтелектуальної навчальної системи

3.2.1. Значення механізму інференції для ефективності навчальних систем

Практичний досвід реалізації інтелектуальних навчальних систем (ІНС) свідчить, що їх загальна ефективність безпосередньо залежить від рівня імплементації механізму інференції (двигуна висновку). Даний механізм може базуватися як на системі правил, так і на обмеженнях, що визначає ступінь інтелектуальності автоматизованих навчальних систем (АНС) на наступних рівнях [47]:

- Генерація зворотного зв'язку – забезпечує адаптацію навчального процесу на основі аналізу помилок, прогалин у знаннях та рівня засвоєння матеріалу студентами.

- Генерація коригувальних навчальних інструкцій – дозволяє модифікувати навчальні завдання та рекомендації відповідно до індивідуальних потреб користувача.

- Інтеграція баз даних, баз знань та сховищ даних ІНС – забезпечує цілісність та узгодженість навчального контенту, а також дозволяє ефективно використовувати накопичені знання та дані для подальшого покращення системи.

Однією з ключових переваг застосування ІНС є можливість моделювання та впровадження комп'ютерного аналога експертної поведінки

викладачів. Це, своєю чергою, дозволяє створювати адаптивні, інтелектуальні та мережеві навчальні середовища. Особливо ефективним є застосування ІНС у таких напрямках, як:

- Навчання технічних дисциплін – наприклад, підготовка студентів у галузі програмування, інженерії та інших технологічних спеціальностей.
- Технічні тренінги для персоналу – автоматизоване навчання спеціалістів, що працюють з технологічними процесами, зокрема в промисловості, енергетиці, медицині тощо.

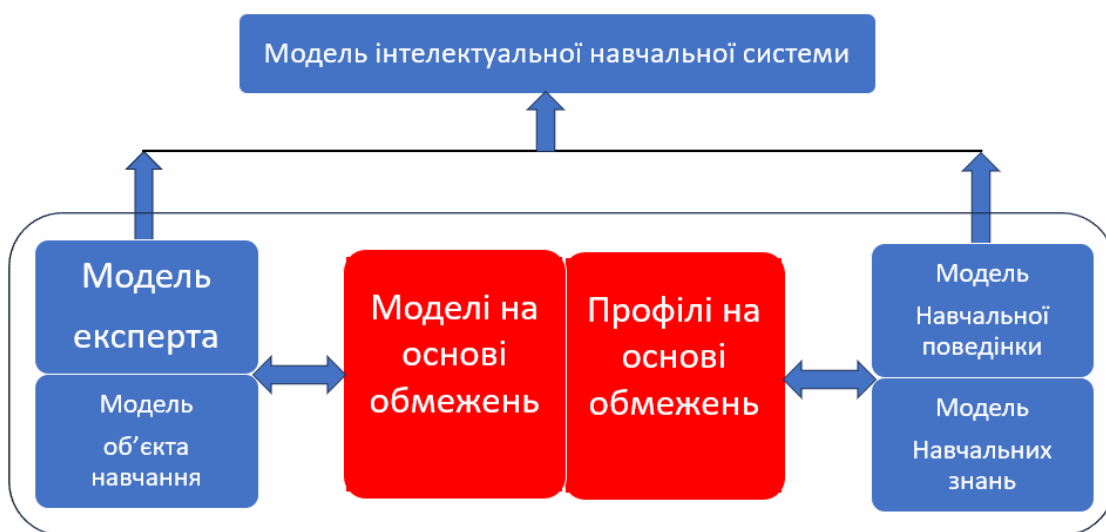


Рис. 3.2. Моделі модулів інтелектуальної навчальної системи

3.2.2. Класифікація користувачів та процес функціонування інтелектуальної навчальної системи

У контексті традиційного підходу до автоматизованих навчальних систем користувачів ІНС поділяють на дві основні категорії [47]:

1. Викладачі (автори курсу, адміністратори);
 2. Об'єкти навчання (студенти, учні, слухачі).
1. Категорія викладачів (автори курсу, адміністратори)

До основних функцій викладачів належать:

- Формування навчального контенту та наповнення системи знаннями предметної області.

- Визначення стратегій навчання, що забезпечують ефективне управління навчальним процесом.

- Розробка тактичних і стратегічних навчальних цілей у вигляді системи взаємопов'язаних задач.

- Адміністрування системи, її технічна підтримка та забезпечення коректного функціонування контенту.

2. Категорія об'єктів навчання (студенти, учні)

Об'єкти навчання поділяються на дві основні групи відповідно до їхньої діяльності:

- Студенти, що займаються навчальною діяльністю – користувачі, які взаємодіють із навчальним контентом у межах освітніх програм, курсів чи тренінгів.

- Учні, що самостійно вивчають різні предмети, наприклад для підготовки до вступу у заклади вищої освіти.



Рис. 3.3. Представлення основних етапів роботи системи

Робота ІНС організована у вигляді множини навчальних сеансів, які проходять через кілька ключових етапів:

1. Ідентифікація поточного стану навчального процесу – аналіз рівня знань, прогресу студента, тощо.

2. Виконання відповідних операцій – для ІНС, що базуються на обмеженнях, цей етап передбачає генерацію множин релевантних, задоволених та порушених обмежень відповідно до стану задачі.

3. Адаптивний вибір подальших дій – механізм обмежень дозволяє аналізувати результати дій студента й автоматично коригувати навчальну траєкторію залежно від успішності засвоєння матеріалу.

4. Перевірка досягнення навчальних цілей – після завершення кожного сеансу здійснюється оцінка того, чи було досягнуто поставлену мету навчання:

- Якщо ціль досягнута, студент отримує системне повідомлення про успішне завершення завдання, а відповідний запис вноситься до його навчального профілю.

- Якщо ціль не досягнута, студент отримує візуалізацію ієрархії обмежень із чіткою індикацією порушених обмежень, що дозволяє йому проаналізувати свої помилки та вдосконалити розуміння матеріалу.

Інтелектуальні навчальні системи забезпечують високу ефективність навчального процесу завдяки інтеграції інтелектуальних механізмів аналізу знань, генерації зворотного зв'язку та автоматизованої корекції навчальних стратегій. Завдяки гнучкості та адаптивності таких систем вони знаходять широке застосування як у традиційній освіті, так і в підготовці різних спеціалістів. Подальший розвиток ІНС пов'язаний із вдосконаленням механізмів інференції, розширенням можливостей персоналізації навчального процесу та інтеграцією з передовими технологіями штучного інтелекту.

3.3. Представлення рівнів інтелектуальності пропонованої системи

Інтелектуальність навчальних систем реалізується на кількох взаємопов'язаних рівнях, кожен з яких відповідає за конкретні аспекти управління навчальним процесом та взаємодії з об'єктами навчання:

Рівень 1: Контроль та супровід об'єктів навчання

Організація доступу об'єктів навчання до навчального контенту та ієрархії навчальних проблем.

Моніторинг дій користувачів у системі та адаптація інтерфейсу відповідно до їхніх потреб.

Впровадження механізмів персоналізованого супроводу навчальної діяльності.

Рівень 2: Ідентифікація та адаптація навчальних стратегій

Первинна реєстрація об'єктів навчання та створення унікальних системних профілів.

Ідентифікація користувачів під час взаємодії з системою на основі аналізу їхніх характеристик та навчальних преференцій.

Динамічне налаштування навчального процесу з урахуванням індивідуальних характеристик кожного об'єкта навчання.

Призначення персоналізованої ієрархії навчальних цілей і підцілей.

Рівень 3: Управління навчальним процесом та адаптивне коригування

Контроль ефективності засвоєння матеріалу відповідно до активованих навчальних стратегій.

Підтримка прийняття рішень шляхом контролю виконання навчальних цілей і підцілей.

Генерація коригувального зворотного зв'язку на основі аналізу успішності навчання.

Надання ієрархічно впорядкованих експертних рекомендацій для корекції навчального процесу.

Основною метою інтелектуальної навчальної системи є реалізація концепції автоматизованого навчання через застосування методів штучного інтелекту. Зокрема, навчальна система інтегрує механізми експертних систем для моделювання діяльності людини-експерта. При цьому ключовим підходом до організації навчального процесу є метод представлення та

задоволення обмежень, що забезпечує адаптивність та гнучкість навчального середовища.

У контексті задач автоматизованого навчання ІНС виконує такі функції:

1. Передача об'єкту навчання певного обсягу знань, що містяться в контентному модулі бази знань.

2. Адаптація навчальних стратегій відповідно до індивідуальних особливостей об'єкта навчання.

3. Інтелектуальний контроль засвоєння знань та корекція навчальних траєкторій.

Оцінка ступеня досягнення навчальних цілей у системі здійснюється шляхом:

- Контролю множини порушених обмежень – аналіз того, наскільки дії об'єкта навчання відповідають встановленим критеріям успішності закладених в системі.

- Аналізу моделі об'єкта навчання – оцінка індивідуального профілю користувача, що містить його навчальні характеристики, рівень засвоєння матеріалу та інші параметри.

- Зіставлення з ідеальною моделлю об'єкта навчання – використання референсної моделі, яка містить типовий набір обмежень, що визначають успішне виконання навчальної мети.

Застосування вказаних підходів дозволяє системі динамічно адаптувати процес навчання, коригувати навчальні стратегії та забезпечувати високий рівень персоналізації навчального середовища.

3.4. Формалізоване представлення моделі об'єкта навчання в навчальній системі

Основним завданням будь-якої інтелектуальної навчальної системи є забезпечення ефективного процесу навчання та вивчення певних розділів

навчального матеріалу. Відповідно до класичної архітектури, ІНС складається з таких основних компонентів [47]:

- Інтерфейс користувача – забезпечує взаємодію суб'єкта навчання із системою, дозволяючи йому отримувати навчальні матеріали, вирішувати завдання та отримувати зворотний зв'язок.

- Педагогічний модуль – визначає часові характеристики та зміст педагогічних дій, здійснює управління навчальними сценаріями та адаптацію процесу навчання відповідно до моделі студента.

- Блок моделювання студента – аналізує рішення, запропоновані суб'єктом навчання, визначає наявні помилки та оновлює профіль студента.

Система містить як теоретичні розділи, так і множину навчальних проблем разом із відповідними еталонними (ідеальними) розв'язками. Деякі тьютори включають доменні модулі, що забезпечують специфічні предметні знання, тоді як інші системи можуть функціонувати без таких модулів, зберігаючи предметну незалежність.

Основним завданням ІНС на кожному етапі є перевірка коректності запропонованих студентом розв'язків. Для цього система застосовує доменні знання, представлені у формі обмежень.

1. Вибір навчальної проблеми:

- На початку навчальної сесії педагогічний модуль обирає конкретну проблему, яку повинен вирішити студент.

- Система може пропонувати проблему або дозволяти студентові самостійно обрати завдання.

2. Аналіз рішення студента:

- Після введення розв'язку студентом педагогічний модуль передає його до блоку моделювання студента.

- Відбувається автоматизований аналіз запропонованого рішення, виявлення помилок та оновлення моделі студента.

3. Генерація адаптивного зворотного зв'язку:

- На основі оновленої моделі студента педагогічний модуль

генерує відповідну педагогічну дію, яка може включати:

- пояснення помилок;
- рекомендації щодо виправлення;
- інструкції щодо подальших навчальних кроків.

4. Перехід до наступної проблеми:

Після вирішення поточної проблеми або за запитом студента педагогічний модуль визначає наступну навчальну задачу, враховуючи поточний профіль студента.

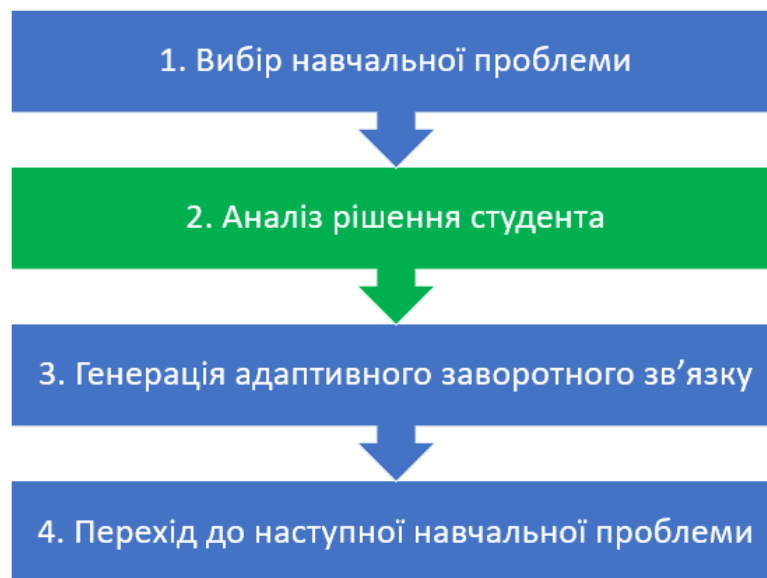


Рис. 3.4. Алгоритмічна схема роботи навчальної системи на основі обмежень

3.4.1. Вимоги до реалізації інтерфейсу системи

Для забезпечення ефективної взаємодії між студентом та ІНС інтерфейс повинен відповідати таким критеріям [49]:

- Робастність (стійкість до помилок) – система має коректно обробляти некоректні дії користувача та надавати зрозумілий зворотний зв'язок.

- Гнучкість – адаптація інтерфейсу відповідно до навчальних потреб студента та його рівня підготовки.

- Простота використання – інтуїтивне управління та доступність

основних функцій для користувача.

Інтерфейс повинен містити такі ключові елементи:

- текст навчальної проблеми;
- структуру запиту для введення рішення;
- пояснення основних концептів предметної області.

Особливу роль у навчальному процесі відіграє педагогічний агент, що виконує функції мотиваційної підтримки та зниження рівня стресу у студентів. Такий агент може бути реалізований у вигляді інтерактивного персонажа (аналогічно до віртуального помічника Windows), який надає корисні підказки та стимулює навчальну активність.

Після розробки ІНС необхідно провести тестування системи та порівняльний аналіз її ефективності з традиційними методами навчання. Дослідження показують, що використання навчальних систем, заснованих на методі обмежень, сприяє більш ефективному засвоєнню матеріалу. Студенти швидше адаптуються до технології навчання, що базується на зворотному зв'язку та автоматизованому контролі.

ІНС може підтримувати кілька варіантів вибору навчальних проблем:

- Послідовний вибір за рівнем складності – студент обирає завдання поступово, переходячи до складніших проблем.
- Автоматичний вибір системою – проблема визначається на основі аналізу профілю студента та його попередніх рішень.
- Вибір на основі математичних моделей – застосування адаптивних алгоритмів, таких як байєсівські мережі, для визначення оптимальної навчальної траєкторії.

3.4.2. Роль обмежень у навчальному процесі

Навчальні проблеми пов'язані з множиною обмежень, які в більшості випадків розподіляються рівномірно. Проте існують два типи обмежень [43]:

- Універсальні обмеження, релевантні для всіх навчальних завдань.
- Специфічні обмеження, характерні лише для певних проблем або

навчальних розділів.

Через постійне оновлення веб-орієнтованих навчальних курсів деякі обмеження можуть втрачати актуальність, що вимагає їх періодичного перегляду.

Перед початком вирішення навчальних проблем система може надавати короткий інструктаж щодо користування ІНС, а також пояснення основних теоретичних положень, необхідних для виконання конкретних завдань [41, 45].

Розглянемо певну БД навчальних проблем інтелектуальної системи у вигляді певної множини навчальних проблем $P_{set} = \{P_i\}, i = 1..n_1$, де кожна проблема P_i має j - станів [47].

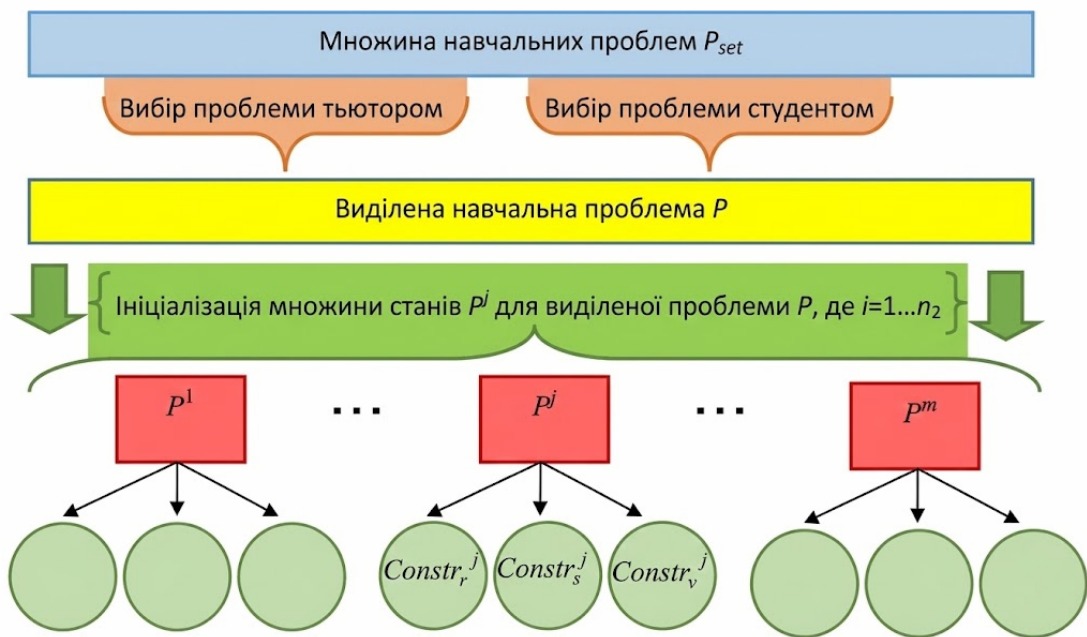


Рис. 3.5. Модель елементів архітектури навчальної системи на основі концепції CSP

Обмеження, що накладається на стан проблеми, визначається як впорядкований триплет $[Constr_r^j, Constr_s^j, Constr_v^j]$, де $Constr_r^j$ є релевантним обмеженням з множини накладених обмежень на поточний стан проблеми; $Constr_s^j$ визначає кількість обмежень, які задовольняються

множиною рішень навчальних проблем; $Constr_r^j$ позначає обмеження, що було порушене в межах поточної сесії.

Представлення знань предметної області здійснюється у вигляді множини обмежень стану проблеми, тобто обмежень, які визначають множину еквівалентних станів проблеми, S де $S=\{s1,s2,\dots,sn\}$. Кожен з отриманих класів еквівалентності ініціює в системі запуск відповідних зворотніх дій, зокрема:

- Індикація правильності або помилковості розв'язку завдання.
- Повідомлення про наявність помилки.
- Генерування підказки.
- Надання детальної підказки.
- Виявлення всіх помилок у розв'язку.
- Відображення правильного рішення.
- Демонстрація порушених обмежень.
- Демонстрація задоволених обмежень.
- Відображення релевантних обмежень.

Кожне обмеження в системі відображає певну властивість предметної області, яка повинна виконуватись для всіх коректних рішень. Якщо обмеження $Constr_r^j$ виконується у певному стані проблеми, то для того, щоб цей стан вважався коректно-релевантним, необхідним є також виконання обмеження $Constr_s^j$. Це означає, що запропоноване студентом рішення $Solution_{set}^{Correct}$ повинно відповідати встановленим критеріям і враховувати поточний стан проблеми.

Досягнення високої ефективності навчальної системи та реалізація її інтелектуальних функцій є неможливими без механізму адаптивного контролю рівня знань студентів (як явного, так і латентного) та коригування дій системи відповідно до індивідуальних здібностей і навчальних потреб користувачів.

Інтелектуальна навчальна система складається зі статичних та

динамічних компонентів. До статичних компонентів належать:

- база даних навчальних проблем,
- база даних рішень,
- база даних обмежень.

До динамічних компонентів належить множина моделей студента, що визначає вибір педагогічного впливу. Процес вибору відповідних дій є ключовим у функціонуванні системи та становить основу процесу моделювання на основі обмежень (Constraint-Based Modeling, CBM).

Розробка та впровадження інтелектуальних тьюторних систем, що базуються на методі обмежень, дозволяє суттєво покращити якість навчального процесу. Використання інтелектуальних механізмів аналізу знань, адаптації навчальних стратегій та автоматизованої генерації зворотного зв'язку забезпечує високий рівень персоналізації навчання, підвищує ефективність засвоєння матеріалу та сприяє формуванню більш гнучкого й адаптивного навчального середовища.

3.5. Реалізація функціональності інтелектуальної навчальної системи

Розроблена інтелектуальна навчальна система є програмним комплексом, призначеним для забезпечення інтелектуальної підтримки процесу вивчення мов програмування [47, 49].

У рамках проєктування системи розроблено профіль 'Вивчення мови C', створений на основі експертних знань та досвіду викладачів, що мають багаторічний досвід викладання курсу 'Програмування мовою C'. Таким чином, розроблена система сприяє:

- накопиченню знань, умінь і навичок у контексті професійної, навчальної та виробничої діяльності;
- організації освітнього процесу із використанням інтелектуальних методів навчання;

- ефективному контролю та адаптації навчального матеріалу відповідно до потреб суб'єкта навчання.

Розроблена система відзначається такими ключовими особливостями:

- Використання формалізованих множин, систем та ієрархій обмежень для моделювання інтелектуальних функцій навчальної системи.

- Застосування формальних правил оперування множинами, системами та ієрархіями обмежень, що забезпечує когнітивну складову адаптації навчального процесу.

- Впровадження методології експертних систем, які у процесі функціонування аналізують та інтерпретують коди або описи програмних помилок, що є наслідком помилкових дій суб'єкта навчання.

Структуризація навчального контенту відповідно до ієрархічної моделі:

$$H = (C, P, S),$$

де

C – множина навчальних курсів,

P – множина проблем, релевантних до певного курсу,

S – множина станів, що описують конкретні навчальні проблеми.

Побудова бази знань системи із таких структурних складових:

- множина помилок E;

- множина обмежень R;

- множина правил П.

Формально множина всіх обмежень у базі знань визначається як:

$$R = \cup_i R_i,$$

де R_i – окремі множини обмежень, що можуть утворювати системи та ієрархії обмежень.

З точки зору структурної організації ця множина утворює послідовність включень:

$$R1 \subseteq R2 \subseteq \dots \subseteq Rn.$$

Веб-орієнтованість та інтерактивність системи, що дозволяє забезпечити доступність та гнучкість навчального процесу. Реалізація архітектури інтелектуальної системи у вигляді доменно-незалежної оболонки, аналогічної до технології експертних систем.

Підтримка інтелектуальної стратегії навчання (або технологічного тренінгу) відповідно до концепції зони найближчого розвитку ZPD (Zone of Proximal Development).

Викладачі, що здійснюють викладання мов програмування, наповнюють базу знань системи відповідно до деревоподібної вкладеної структури:

$$C \supset P \supset S,$$

де:

C – множина навчальних курсів,

P – множина проблем, пов'язаних із певним курсом,

S – множина станів, що характеризують конкретну навчальну проблему.

На рівні навчальних проблем та їх станів застосовуються відповідні множини обмежень, які регламентують допустимі та недопустимі рішення в межах навчального процесу.

До сформованих обмежень додають типові помилки характерні для даного навчального курсу.

Також до обмежень додають 4 рівні експертних порад для користувачів різних рівнів. Таким чином формується база знань системи за допомогою якої відбуватиметься процес навчання.

Система використовується при вивчення мов програмування, в ході якого виникає ряд навчальних проблем, які описуються відповідними помилками компілятора та інтерпретатора. Вводячи отримані помилки через інтерфейс користувача, система ідентифікує помилку, видає перелік

порушених обмежень, що призвели до неї і до кожного з таких обмежень генерує перелік порад різних рівнів у порядку спадання складності та релевантності пояснень (від найвищого до найнижчого). Таким чином користувач зможе отримати компетентну експертну пораду при вирішенні навчальної проблеми, відповідно до його рівня знань (зони найближчого розвитку) [36, 37].

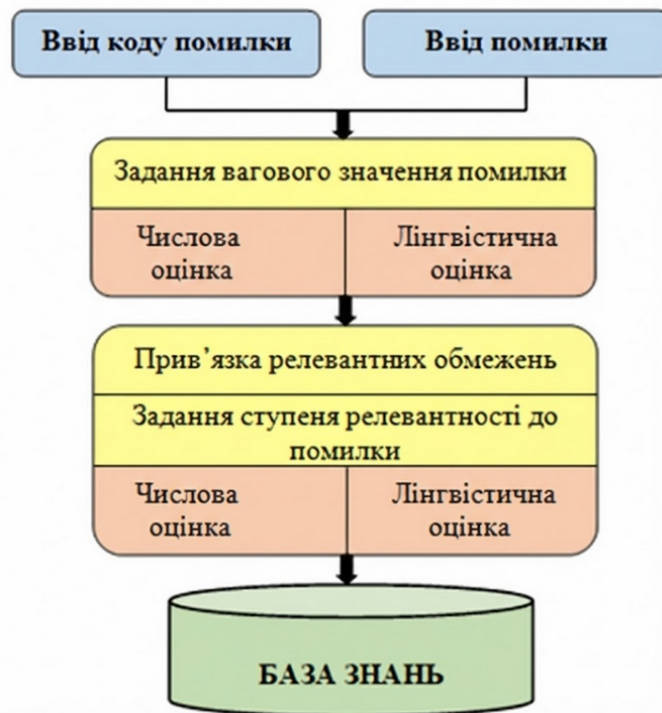


Рис. 3.6. Додавання помилок до бази знань

Розглянемо інтерфейс та основні функціональні можливості запропонованої системи.

Рис. 3.7. Форма входу в систему

ProgLang

Ви зайшли як **admin admin** [Вийти](#)

[Додати користувача](#)
[Перелік користувачів](#)
[Додати навчальний курс/проблему/стан](#)
[Додати обмеження/помилку](#)
[Редагувати обмеження](#)
[Видалення помилок](#)
[Статистика](#)
[На головну](#)

Рис. 3.8. Інтерфейс головної сторінки системи

Основними структурними одиницями для наповнення системи є навчальні курси, навчальні проблеми, обмеження, типові помилки, експертні поради. Модуль додавання навчального курсу (рис. 3.9) містить три пункти: навчальний курс, навчальна проблема, стан навчальної проблеми. Для кожного вибору забезпечено додавання, видалення, коректування та вибір існуючих даних.

Навчальний курс	
Перелік навчальних курсів:	<input type="text" value="C"/> змінити
Новий курс	<input type="text"/>
+ Додати ще один курс	
<input type="button" value="Скинути"/>	<input type="button" value="Додати"/>
Навчальна проблема	
Перелік навчальних проблем вибраного курсу:	<input type="text" value="Функції"/> змінити
Нова навчальна проблема	<input type="text"/>
+ Додати ще одну проблему	
<input type="button" value="Скинути"/>	<input type="button" value="Додати"/>
Стан навчальної проблеми	
Перелік станів навчальних проблем вибраного курсу:	<input type="text" value="Структура"/> змінити
Стан навчальної проблеми	<input type="text"/>
+ Додати ще один стан	
<input type="button" value="Скинути"/>	<input type="button" value="Додати"/>

Рис. 3.9. Вікно додавання нового курсу/проблеми/стану

Даний модуль дозволяє здійснювати внесення в базу даних навчальних курсів, проблем навчальних курсів та станів проблем.

Б1. — Блок опису обмежень до поточного стану поточної навчальної проблеми

Навчальний курс з програмування:

Навчальна проблема:

Додати стан навчальної проблеми

Присвоєння

Обмеження

Вводити українською мовою, наприклад: Всі оператори повинні закінчуватися ";"

Релевантність обмеження до стану проблеми ?	<input type="text" value="74"/>	Лінгвістична оцінка	<input type="text"/>
Ваговий коефіцієнт обмеження (вага присвоєння) ?	<input type="text" value="74"/>	Лінгвістична оцінка	<input type="text"/>
Ймовірнісний коефіцієнт (ймовірність задоволення обмеження) ?	<input type="text" value="74"/>	Лінгвістична оцінка	<input type="text"/>
Можливісний коефіцієнт (можливість задоволення обмеження) ?	<input type="text" value="74"/>	Лінгвістична оцінка	<input type="text"/>
Преференцію обмеження (першочерговість задоволення) ?	<input type="text" value="74"/>	Лінгвістична оцінка	<input type="text"/>

Експертні поради тьютора

Початковий рівень

```
int main() {
    int a;
    1 = a; // C2106
}
```

Спробуйте так: `a = 1;`

Задовільний рівень

Лівий операнд виразу хибний.
Спробуйте поміняти правий і лівий операнд місцями.

Хороший рівень

Невірний тип операнда

Відмінний рівень

Помилка присвоєння

+ Додати ще одне обмеження

Рис. 3.10. Додавання нового обмеження

Модуль “додавання обмежень і помилок” поділяється на два розділи. В першому можна додавати обмеження релевантні до навчальної проблеми або

стану навчальної проблеми (рис. 3.10). Якщо обмеження релевантне до проблеми, стан навчальної проблеми не активується і відповідно при внесенні даних не враховується. Всі поля є обов'язковими для заповнення. Також необхідно вказати релевантність обмеження до проблеми, ваговий коефіцієнт обмеження, ймовірнісний коефіцієнт обмеження, можливісний коефіцієнт обмеження та преференцію обмеження за допомогою бігунка у вигляді числового значення або за допомогою списку вибору для лінгвістичних значень оцінки.

Експертні поради генеруються для різного рівня знань об'єктів навчання, починаючи з найнижчого – деталізований опис способу вирішення навчальної проблеми, яка може скластися у випадку порушення даного обмеження і до найвищого – в якому тільки вказується на те в чому проблема без деталізованого опису.

Модуль редагування обмежень (рис. 3.11) за інтерфейсом схожий до інтерфейсу додавання обмежень.

Редагування обмежень

Навчальний курс з програмування:

Навчальна проблема:

Стан навчальної проблеми:

Обмеження [змінити](#)

[редагувати](#) [видалити](#)

Експертні поради тьютора

Помилка при відкритті тіла функції. Здійснено подвійне відкриття тіла функції. Щоб усунути помилку видаліть одна з двох відкриттів.

Початковий рівень
 Приклад помилки:

```
void Func(int);
void Func(int) // переше відкриття
```

При написанні функції двічі використаний оператор відкриття тіла функції. Щоб усунути помилку видаліть зайве відкриття.

Задовільний рівень

Хороший рівень

Відмінний рівень

Подвійне відкриття тіла функції.

Структурна помилка при відкритті тіла функції.

Рис. 3.11. Модуль редагування обмеження

Видалення помилок відбувається наступним чином: із випадуючого списку переліку всіх помилок вибирається необхідна помилка і після цього необхідно натиснути на кнопку “видалити”. Відповідно дана помилка буде видалена і всі зв’язки пов’язані з нею також будуть знищені. У випадку, якщо не вибрана жодна помилка, кнопка “видали” не буде виконувати ніяких дій при натисненні на неї.

Розглянемо інтерфейс системи додавання навчального курсу, навчальної проблеми та стану навчальної проблеми.

Ви зайшли як **admin admin** [Вийти](#)

[повернутися](#)

Навчальний курс

Прелік навчальних курсів: Perl [змінити](#)

Новий курс

x C
x PHP
+ Додати ще один курс

Рис. 3.12. Додавання навчального курсу

Ви зайшли як **admin admin** [Вийти](#)

[повернутися](#)

Навчальний курс

Прелік навчальних курсів: Perl [змінити](#)

Новий курс

x C
x PHP
+ Додати ще один курс

- C
- Java
- PHP
- JavaScript
- Basic
- c++
- Perl

Рис. 3.13. Результат додавання навчального курсу

Ви зайшли як **admin admin** [Вийти](#)

[повернутися](#)

Навчальна проблема

Перелік навчальних проблем вибраного курсу: - [змінити](#)

Нова навчальна проблема

Типи даних

x Оператори
x Масиви
x Функції
+ Додати ще одну проблему

Рис. 3.14. Додавання навчальних проблем для курсу C

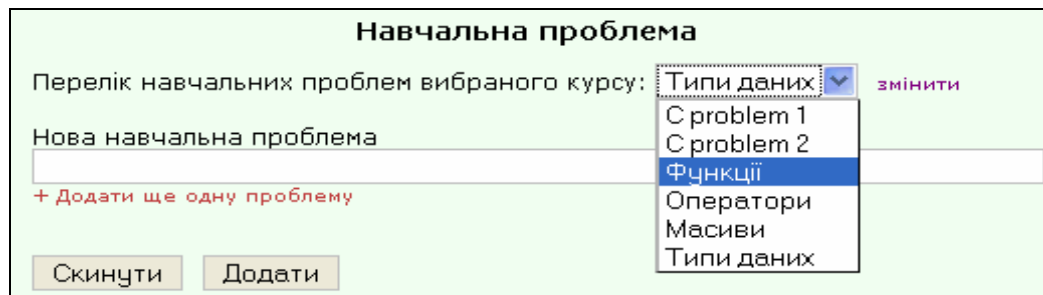


Рис. 3.15. Результат додавання навчальних проблем

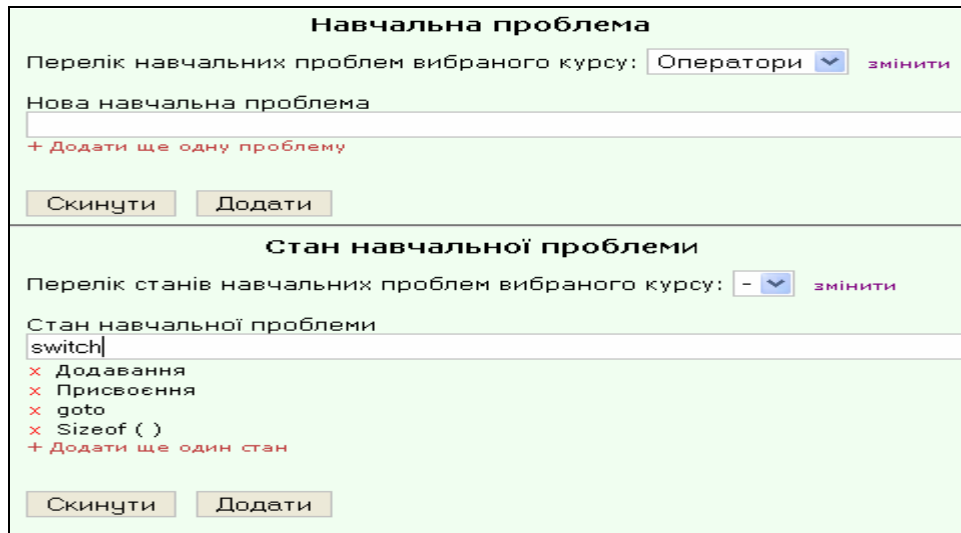


Рис. 3.16. Додавання стану для навчальної проблеми 'Оператори'

Модуль статистики (рис. 3.17) відображає інформацію про користувачів системи, а саме, веде облік порушених та задоволених обмежень користувачем.

Для того, щоб отримати необхідну інформацію, потрібно послідовно заповнити даними запропоновані випадваючі списки. Кожен попередній список підвантажує інформацію для наступного, якщо вона є в базі.

Перш за все вибирається тип користувача, якщо вибрано "студента", тоді з'являється перелік груп студентів, у випадку профілю "викладач" та "розробник" цей список є відсутній. Для студента вибирається ще група, потім вибирається його прізвище з списку зареєстрованих користувачів (списку групи).

Після виконання запиту висвітлюється кількість сесій даного користувача і їх перелік. Після вибору певної сесії відображається кількість

курсів, які пройшов користувач і їх перелік. Після вибору навчального курсу пройденого користувачем висвітлюється кількість проблем порушених у даному курсі користувачем, а також їхній перелік. Після вибору навчальної проблеми порушеної користувачем висвітлюється кількість станів проблем у даному курсі їхній перелік, а також підраховується кількість релевантних, задоволених і порушених обмежень для даної проблеми, оскільки стану проблема може не містити.

Статистика

Тип користувача: студент

Група: ПЗ

Перелік користувачів: test S test N

Загальна кількість сесій: 8

Сесії: Session_8

Загальна кількість пройдених навчальних курсів: 1

Навчальні курси: C

Загальна кількість порушених навчальних проблем: 1

Навчальні проблеми: Оператори

Загальна кількість порушених станів навчальних проблем: 1

Стани навчальних проблем: Присвоєння

Релевантні обмеження: 1

Задоволені обмеження: 0

Порушені обмеження: 1

Статистика по навчальних курсах

Для активації списку слід вибрати користувача

Перелік навчальних курсів: C

Релевантні обмеження: 16

Задоволені обмеження: 13

Порушені обмеження: 3

Рис. 3.17. Загальний вигляд вікна статистики

Вибираючи певний стан проблеми, отримуються дані про релевантні, задоволені, порушені обмеження для стану проблеми. По кожному з користувачів існує можливість перегляду статистичних даних по обмеженнях

для вибраного навчального курсу. Дана статистична інформація доступна тільки у випадку вибраного користувача.



Рис. 3.18. Алгоритмічна схема формування статистики порушених та задоволених обмежень

Формування статистичних оцінок щодо порушених R^- та задоволених R^+ обмежень здійснюється за наступним алгоритмом.

1. Вибір користувача.

З множини зареєстрованих користувачів системи вибирається конкретний користувач U_i , для якого необхідно отримати статистичні дані.

2. Фіксація сесійної активності.

Для кожного користувача ведеться облік входжень у систему та реєстрація відповідних сесій S_j . У межах кожної сесії записується інформація про дії, виконані користувачем, включаючи порушення певних обмежень.

3. Аналіз порушених обмежень на рівні навчальних курсів.

Після вибору конкретної сесії S_j для вибраного користувача U_i визначається множина навчальних курсів S_k , обмеження яких були порушені в межах даної сесії.

4. Аналіз порушених обмежень на рівні навчальних проблем.

Якщо користувачем був обраний конкретний навчальний курс S_k , формується множина навчальних проблем P_m , обмеження яких були порушені під час даної сесії.

5. Аналіз станів навчальних проблем.

Після вибору конкретної навчальної проблеми P_m визначається множина її станів S_n , для яких були зафіксовані порушення обмежень у вибраній сесії.

6. Розрахунок статистичних показників.

Для кожної навчальної проблеми P_m обчислюються:

- загальна кількість релевантних обмежень R^* ,
- кількість порушених обмежень R^- ,
- кількість задоволених обмежень R^+ .

У разі вибору конкретного стану навчальної проблеми S_n , виконується остаточний підрахунок показників R^* , R^- , R^+ у межах вибраного курсу S_k для конкретної сесії S_j .

Таким чином, статистичний аналіз дозволяє оцінити ступінь відповідності дій користувача навчальним цілям та сформулювати рекомендації щодо подальшої адаптації навчального процесу відповідно до концепції зони найближчого розвитку.

Висновки до розділу

У третьому розділі проведено аналіз та імплементацію концепцій штучного інтелекту для побудови інтелектуальної навчальної системи. Було розглянуто структурну схему системи, визначено її основні модулі та їхню функціональність, що забезпечує адаптивність навчального процесу.

Описано моделі модулів інтелектуальної навчальної системи, серед яких ключову роль відіграє механізм інференції, який дозволяє реалізувати адаптивні алгоритми навчання та персоналізовані рекомендації для користувачів. Запропоновано класифікацію користувачів, яка сприяє ефективному управлінню навчальними процесами та формуванню індивідуальних навчальних траєкторій.

Особливу увагу приділено представленню рівнів інтелектуальності навчальної системи, що дозволяє оцінювати її здатність до адаптації, аналізу поведінки користувачів та автоматичного коригування навчального контенту відповідно до потреб студентів.

Формалізовано модель об'єкта навчання в навчальній системі, визначено вимоги до реалізації інтерфейсу, що забезпечує зручність взаємодії користувачів із системою. Також було підкреслено важливість обмежень у навчальному процесі, оскільки вони сприяють коректному вибору навчальних матеріалів, тестових завдань та оптимізації освітньої траєкторії.

Реалізація функціональності інтелектуальної навчальної системи продемонструвала можливість ефективного використання технологій штучного інтелекту для персоналізованого навчання, адаптації контенту та автоматизації оцінювання знань.

Таким чином, результати даного розділу підтвердили, що застосування концепцій штучного інтелекту в навчальних системах дозволяє значно підвищити їхню ефективність, адаптивність та якість навчального процесу.

ВИСНОВКИ

У магістерській роботі було проведено дослідження концепцій штучного інтелекту та їх застосування в навчальних системах. Основна увага приділена автоматизованим та інтелектуальним навчальним системам, методам засвоєння знань, використанню обмежень у навчальному процесі та імплементації програмних рішень на основі концепції задоволення обмежень (CSP).

У першому розділі здійснено аналіз сучасних автоматизованих та інтелектуальних навчальних систем. Було розглянуто стратегії засвоєння знань, включаючи біхевіористичний підхід, концепцію представлення знань через обмеження та адаптацію методу Краудера. Особливу увагу приділено моделі контролю навчального процесу та мультимодальній моделі онлайн-навчання, які дозволяють підвищити ефективність взаємодії між студентами та системою.

У другому розділі досліджено методи та алгоритми концепції задоволення обмежень (CSP) у контексті навчальних систем. Було розглянуто формалізоване представлення цієї концепції, що дозволяє моделювати процеси навчання та тестування. Аналізовано застосування CSP для розв'язання таких завдань, як автоматизоване складання розкладу, персоналізоване навчання та автоматична генерація тестів. Реалізація цих підходів сприяє підвищенню ефективності навчального процесу та адаптації освітніх ресурсів до потреб студентів.

Третій розділ присвячено практичній імплементації штучного інтелекту в навчальних системах. Описано структурну схему навчальної системи та представлено моделі її основних модулів. Визначено значення механізму інференції для підвищення ефективності навчання, розглянуто класифікацію користувачів та рівні інтелектуальності системи. Було формалізовано модель об'єкта навчання, визначено вимоги до інтерфейсу та підкреслено роль обмежень у навчальному процесі. Запропоновано

реалізацію функціональності інтелектуальної навчальної системи, що демонструє можливості сучасних технологій штучного інтелекту для персоналізації та автоматизації навчання.

Узагальнюючи результати дослідження, можна зробити висновок, що використання концепцій штучного інтелекту, зокрема CSP, дозволяє значно підвищити ефективність навчальних систем, зробити їх адаптивними до потреб студентів та забезпечити автоматизацію ключових етапів навчального процесу. Отримані результати можуть бути використані для подальшого розвитку інтелектуальних навчальних платформ, що сприятиме підвищенню якості освіти та персоналізації навчального контенту.

Запропонована інтелектуальна навчальна система поєднує методи експертних систем, ієрархічне моделювання навчального контенту та адаптивні підходи до навчання. Її особливості, зокрема підтримка навчання на основі множин обмежень та зворотного зв'язку, сприяють підвищенню ефективності освітнього процесу. Впровадження доменно-незалежної архітектури та веб-орієнтованості забезпечує гнучкість системи та її застосування в різних предметних галузях.

ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Sajja, R., Cwiertny, D., & Demir, I., Sermet, Y., Cikmaz, M., (2023). Artificial intelligence-enabled intelligent assistant for personalized and adaptive learning in higher education. arXiv preprint. <https://arxiv.org/abs/2309.10892>
2. Adamu, S., & Awwalu, J. (2019). The role of artificial intelligence (AI) in adaptive eLearning system (AES) content formation: Risks and opportunities involved. arXiv preprint. <https://arxiv.org/abs/1903.00934>
3. Roman Barták. Constraint programming: a survey of solving technology. *AIRONews*, IV(4):7–11, 1999.
4. Jia, J. (2003). CSIEC (Computer Simulator in Educational Communication): An intelligent web-based teaching system for foreign language learning. arXiv preprint. <https://arxiv.org/abs/cs/0312030>
5. Wróblewska, A., Witas, M., Frańczak, K., Książ, A., Cheong, S. A., Chee, T. S., Hołyst, J., & Paprzycki, M. (2024). Intelligent interface: Enhancing lecture engagement with didactic activity summaries. arXiv preprint. <https://arxiv.org/abs/2406.14266>
6. Graesser, A. C., Chipman, P., Haynes, B. C., & Olney, A. (2005). AutoTutor: An intelligent tutoring system with mixed-initiative dialogue. *IEEE Transactions on Education*, 48(4), 612–618.
7. Mitrovic, A., Ohlsson, S. Constraint-based knowledge representation for individualized instruction. *Artificial Intelligence in Education*, 16 (2), 2006, 135–169.
8. Koedinger, K. R., & Corbett, A. T. (2006). Cognitive tutors: Technology bringing learning sciences to the classroom. In R. K. Sawyer (Ed.), *The Cambridge handbook of the learning sciences* (pp. 61–78). Cambridge University Press.

9. VanLehn, K. (2011). The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist*, 46(4), 197–221. <https://doi.org/10.1080/00461520.2011.6113>
10. T. Anderson, *The theory and practice of online learning*. Athabasca University Press, 2008.
11. VanLehn, K. (2006). The behavior of tutoring systems. *International Journal of Artificial Intelligence in Education*, 16(3), 227–265.
12. Wenger, E. (1987). *Artificial intelligence and tutoring systems: Computational and cognitive approaches to the communication of knowledge*. Morgan Kaufmann.
13. Woolf, B. P. (2009). *Building intelligent interactive tutors: Student-centered strategies for revolutionizing e-learning*. Morgan Kaufmann.
14. Eugene C. Freuder. Synthesizing constraint expressions. *Communications of the ACM*, 21(11):958–966, 1978.
15. Helmut Simonis. Building industrial applications. In *International Summer School on Constraints in Computational Logics*. Esprit Working Group CCL, 1999.
16. Slim Abdennadher and Hans Schlenker. Nurse scheduling using constraint logic programming. In *Proceedings of the 6th National Conference on Artificial Intelligence (AAAI-99); Proceedings of the 11th Conference on Innovative Applications of Artificial Intelligence*, pages 838–843, Menlo Park, Cal., 1999. AAAI/MIT Press.
17. Mark Wallace. Applying constraints for scheduling. In Brian Mayoh, Enn Tyugu, and Jaan Penjam, editors, *Constraint Programming*, volume F: *Computer and Systems Sciences 131 of NATO Advanced Science Institute Series*, pages 153–171. Springer-Verlag, 1992.
18. Pascal Van Hentenryck. *Constraint Satisfaction in Logic Programming*. MIT Press,
19. Zsófia Ruttkay. Constraint satisfaction — a survey. *CWI Quarterly*, 11(2&3):123–161, 1998.

20. Smith Barbara M. A tutorial on constraint programming. Technical Report 95.14, University of Leeds, School of Computer Studies, 1995.
21. Vipin Kumar. Algorithms for constraint satisfaction problems: A survey. *AI Magazine*, 13(1):32–44, 1992.
22. Philippe Baptiste. A Theoretical and Experimental Study of Resource Constraint Propagation. PhD thesis, University of Compiègne, 1998.
23. Wilhelmus P. M. Nuijten. Time and resource constrained scheduling: a constraint satisfaction approach. PhD thesis, Eindhoven University of Technology, 1994.
24. Thomas Schiex. Possibilistic constraint satisfaction problems or “How to handle soft constraints?”. In 8th International Conference on Uncertainty in Artificial Intelligence, pages 268–275, Stanford, CA, July 1992.
25. Zsófia Ruttkay. Fuzzy constraint satisfaction. In Proceedings of the Third IEEE International Conference on Fuzzy Systems, pages 1263–1268. IEEE Press, 1994.
26. Didier Dubois, Hélène Fargier, and Henri Prade. Possibility theory in constraint satisfaction problems: Handling priority, preference and uncertainty. *Applied Intelligence*, 6:287–309, 1996.
27. C. K. Chiu and Jimmy Ho-man Lee. Extending HCLP with partially ordered hierarchies and composite constraints. *Journal of Experimental and Theoretical Artificial Intelligence*, 10:5–24, 1998.
28. Michael Jampel, Jean-Marie Jacquet, David Gilbert, and Sebastian Hunt. Transformation between HCLP and PCSP. In Eugene C. Freuder, editor, *Principles and Practice of Constraint Programming’ – CP96*, pages 237–251. Springer-Verlag LNCS 1118, 1996.
29. Hana Rudová. Over-constrained systems. In Proceedings of the Sixteenth National Conference on Artificial Intelligence and the Eleventh Innovative Applications of Artificial Intelligence Conference, page 954. AAAI Press/MIT Press, 1999. Presented in 1999 SIGART/AAAI Doctoral Consortium.

30. Dechter, R. (2003). *Constraint Processing*. Morgan Kaufmann.
31. Tsang, E. (1993). *Foundations of Constraint Satisfaction*. Academic Press.
32. Mackworth, A. K. (1977). Consistency in networks of relations. *Artificial Intelligence*, 8(1), 99-118.
33. Freuder, E. C. (1982). A sufficient condition for backtrack-free search. *Journal of the ACM*, 29(1), 24-32.
34. Montanari, U. (1974). Networks of constraints: Fundamental properties and applications to picture processing. *Information Sciences*, 7, 95-132.
35. Rossi, F., van Beek, P., & Walsh, T. (Eds.). (2006). *Handbook of Constraint Programming*. Elsevier.
36. Bessière, C. (2006). Constraint propagation. In F. Rossi, P. van Beek, & T. Walsh (Eds.), *Handbook of Constraint Programming* (pp. 29-83). Elsevier.
37. Garey, M. R., & Johnson, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman.
38. Schrijver, A. (2003). *Combinatorial Optimization: Polyhedra and Efficiency*. Springer.
39. Bulatov, A. A. (2017). A dichotomy theorem for nonuniform CSPs. In *Proceedings of the 58th Annual IEEE Symposium on Foundations of Computer Science (FOCS)* (pp. 319-330). IEEE.
40. Jeavons, P., Cohen, D., & Gyssens, M. (1997). Closure properties of constraints. *Journal of the ACM*, 44(4), 527-548.
41. Вовк Р.Б., Криль Л.Р., Королик В.М. Дослідження структури зворотного зв'язку тьюторної системи на основі обмежень // *Радиоэлектроника и молодежь в XXI веке: Материалы XIII Международного молодежного форума (30 марта – 1 апреля 2009 г., Харьков)*. – Х.: ХНУРЭ., 2009. – ч.2 – С. 167.
42. Cohen, D. A., Cooper, M. C., Jeavons, P. G., & Krokhin, A. A. (2006). A maximal tractable class of soft constraints. *Journal of Artificial Intelligence Research*, 22, 1-22.

43. Kumar, V. (1992). Algorithms for constraint-satisfaction problems: A survey. *AI Magazine*, 13(1), 32-44.
44. Bacchus, F., & Grove, A. (1995). On the forward checking algorithm. In *Proceedings of the 6th International Conference on Principles of Knowledge Representation and Reasoning (KR)* (pp. 559-565). Morgan Kaufmann.
45. Вовк Р.Б., Шекета В.І. Формалізоване представлення моделі тьюторної системи на основі концепції задоволення обмежень // Матеріали ІХ міжнародної конференції “ Інтелектуальний аналіз інформації ІАІ-2009”, – м. Київ, 19-22 травня 2009. – С. 48 – 54.
46. Marriott, K., & Stuckey, P. J. (1998). *Programming with Constraints: An Introduction*. MIT Press.
47. Вовк Р.Б., Шекета В.І. Формалізація сумарної моделі студента в інтелектуальній тьюторній системі на основі обмежень // Вісник Хмельницького національного університету. Технічні науки – 2009., том 2., – с. 242 – 248.
48. Bulatov, A. A., & Marx, D. (2012). Constraint satisfaction parameterized by solution size. In *Proceedings of the 44th Annual ACM Symposium on Theory of Computing (STOC)* (pp. 687-694). ACM.
49. Вовк Р.Б. Реалізація інтерфейсних елементів інтелектуальної тьюторної системи на основі обмежень // Вісник Кременчуцького державного університету імені Михайла Остроградського 2010., Вип. 1/2010 (60) частина 1., с. 39 – 43.
50. Вовк Р.Б., Шекета В.І., та ін. Розробка методу інтеграції машинного навчання з виведенням на основі правил // DOI: <https://doi.org/10.31891/2307-5732-2025-351-12>.

ДОДАТКИ

Додаток А

Фрагмент коду розробленої навчальної системи на основі концепції CSP

index.htm

```
<html>
<head><title>ProgLang:: перша сторінка</title>
<div class='b'>Вибір помилки</div><br/>
<form action='' method='POST'>
<div><span>Введіть помилку </span><input name='name' type='text' size='40' /></div>
<div><input class='reset' type='reset' value='Скинути' /><input type='submit'
value='Відправити' /></div>
</form>
<?php
    if ($_SERVER['REQUEST_METHOD'] == 'POST') {
        echo '<h1>, <b>' . $_POST['name'] . '</b></h1>!';
    }
?>
</body>
</html>
```

login.php

```
<?php
    session_start();
    if ($_SERVER['REQUEST_METHOD'] == 'POST') {
        $Host = 'localhost';
        $User = 'root';
        $Password = '';
        $DBName = 'mysql';
        $Link = mysql_connect($Host, $User, $Password);
        mysql_select_db($DBName, $Link);
        $login = $_POST['login'];
        $pass = $_POST['pass'];
        // аутентифікація користувача
        mysql_query('SELECT `users`.`surname` FROM `users`
                    WHERE `users`.`login` =
`$login`');
        // перевірка правильності логіна
        if(mysql_affected_rows() != 0){
            $result = mysql_query('SELECT `users`.`id`, `users`.`name`,
`users`.`surname`, `users`.`usertype` FROM `users`
                    WHERE `users`.`login` = '$login'
AND `users`.`password` = '$pass`');
            // перевірка правильності пароля
            if(mysql_affected_rows() != 0){
                $row = mysql_fetch_array($result);
                $_SESSION['username'] = $login;
                $_SESSION['name'] = $row['name'];
                $_SESSION['surname'] = $row['surname'];
                $_SESSION['right'] = $row['usertype'];
                $gd = getdate();
                // формування часу входу в систему
                $dt = strval($gd['year']).'-' . strval($gd['mon']).'-'
                . strval($gd['mday']).'
                . strval($gd['hours']).':' . strval($gd['minutes']).':' . strval($gd['seconds']);
                $id = $row['id'];
                // занесення даних про залогінення користувача
                $res = mysql_fetch_array(mysql_query('SELECT
count(`session`.`id_user`) AS `howmany` FROM `session` WHERE `session`.`id_user` =
'$id`'));
                $Snum = $res['howmany'] + 1;
                $Sname = 'Session_' . strval($Snum);
                mysql_query('INSERT INTO `session`
                    VALUES('0', '$id', '$Sname', '$dt)');
                $res = mysql_fetch_array(mysql_query('SELECT `session`.`id`
FROM `session`
                    WHERE `session`.`id_user` = '$id'
AND `session`.`name` = '$Sname`'));
            }
        }
    }
}
```

```

        $_SESSION['idses'] = $res['id'];
    }
    else{
        echo '<div class='Elog'>Не вірний пароль</div>';
    }
}
else{
    echo '<div class='Elog'>Користувача з логіном
<strong>'. $login.'</strong> не існує</div>';
}
}
// підключення скрипта виходу з системи
include 'logout.php';
?>
<html>
<head><title> ProgLang :: Вхід в систему</title>
</head>
<body>
<form name='loginform' method='post'>
<input id='out' name='out' type='hidden' value='' />
<?php
    if(isset($_SESSION['username'])) echo '<div style='display:none'>';
?>
<center>
<div class='logform'>
<table>
<tr><td><span>Логін </span></td><td><input id='login' name='login' type='text'
size='20' /></td></tr>
<tr><td><span>Пароль </span></td><td><input id='pass' name='pass' type='password'
size='20' /></td></tr>
</table>

<br/><div><input class='reset' type='reset' value='Скинути' /><input id='enterBtn'
disabled='disabled' type='submit' value='Ввійти' /></div>
</div>
</center>
<?php
    if(isset($_SESSION['username'])) echo '</div>';
?>
</form>
<div class='ml10 mb5'><a href='index.php'>На головну</a></div>
<?php
    if(isset($_SESSION['username']) && $_SESSION['right']>1) {
        echo '<div class='ml10'><a href='adminka.php'>Адміністративний
модуль</a></div>';
    }
?>
<?php
    include 'bottom.php';
?>
</body>
<script type='text/javascript'>var g_forInitMas =
'enterBtn[inp],login[k],pass[k]';</script>
<script>
// активація/деактивація кнопки входу в систему
m_login.ku = m_pass.ku = function(){
    if(m_login.value.length > 0 && m_pass.value.length > 0){
        m_enterBtn.en()
    }
    else{
        m_enterBtn.dis()
    }
}
</script>
</html>

<?php
    session_start();
?>
<html>
<head><title>ProgLang :: Адміністративний модуль</title>

```

```

</head>
<body>

adminka.php
<?php
    //session_start();
    include 'logout.php';
    echo '<div class='ml10'>';
    if(isset($_SESSION['username'])) {
        if($_SESSION['right']>2){
            echo '<div class='mb5'><a href='newuser.php'>Додати
користувача</a></div>';
        }
        if($_SESSION['right']>1){
            echo '<div class='mb5'><a href='userList.php'>Перелік
користувачів</a></div>';
            echo '<div class='mb5'><a href='newCPS.php'>Додати навчальний
курс/проблему/стан</a></div>';
            echo '<div class='mb5'><a href='newCon.php'>Додати
обмеження/помилку</a></div>';
            echo '<div class='mb5'><a href='editCon.php'>Редагувати
обмеження</a></div>';
            echo '<div class='mb5'><a href='delError.php'>Видалення
помилки</a></div>';
            echo '<div class='mb5'><a href='statistics.php'>Статистика</a></div>';
        }
    }

?>
<div><a href='index.php'>На головну</a></div>
</div>
<?php
    include 'bottom.php';
?>
</body>
</html>

```

```

change_con.php
<?php
$prob = $_GET['q'];
$Host = 'localhost';
$User = 'root';
>Password = '';
$DBName = 'mysql';
$Link = mysql_connect($Host, $User, $Password);
mysql_select_db($DBName, $Link);
$arr = array();
$arr = preg_split('/\|/', $prob);
if(count($arr)==3){
    $result = mysql_query('SELECT `constraints`.`id`, `constraints`.`name` FROM
`constraints`
WHERE `constraints`.`id_lang`='$arr[0]' AND
`constraints`.`id_prob`='$arr[1]' AND `constraints`.`id_state`='$arr[2]');
}
else{
    $result = mysql_query('SELECT `constraints`.`id`, `constraints`.`name` FROM
`constraints`
WHERE `constraints`.`id_lang`='$arr[0]' AND
`constraints`.`id_prob`='$arr[1]');
}
while($row = mysql_fetch_array($result)){
    if (!$srow){
        $srow = $row['id'].'^'.$row['name'];
    }
    else{
        $srow = $srow.'|'.$row['id'].'^'.$row['name'];
    }
}
echo $srow;
mysql_close($Link);

```

```
?>
```

error_find.php

```
<html>
<head><title>Пошук помилки</title>
<link rel='stylesheet' type='text/css' href='main.css' />
</head>
<body>
<?php
$name = $_POST[name];
$Host = 'localhost';
>User = 'root';
>Password = '';
>DBName = 'mysql';
>Link = mysql_connect($Host, $User, $Password);
mysql_select_db($DBName, $Link);
$result = mysql_query('SELECT errors.name FROM errors
                        WHERE MATCH(errors.name) AGAINST ('.$name' IN
BOOLEAN MODE)');
echo '<div>Співпадання по слову `.$name.`</div>';
while($row = mysql_fetch_array($result)){
    echo '<div>`.$row['name'].`</div>';
}
mysql_close($Link);
?>
<div><a href='index.php'>назад</a></div>
</body>
</html>
```

newCPS.php

```
<?php
    session_start();
    if(!isset($_SESSION['username']) || $_SESSION['right']<2){
        echo '<div>У вас немає прав доступу до сторінки або ви не <a
href='login.php'>увійшли в систему</a></div>';
        exit;
    }
?>
<html>
<head><title>Адміністративний модуль :: Додати навчальний курс</title>
<meta http-equiv='content-Type' content='text/html; charset=windows-1251' />
<link rel='stylesheet' type='text/css' href='css/main.css' />
<link rel='stylesheet' type='text/css' href='css/lib.css' />
</head>
<body>
<?php
    include 'logout.php';
?>
<div class='ml10'><a href='adminka.php'>повернутися</a></div><br/>
<form method='post' name='addForm' action='newCPS_serv.php'>
<h1>Навчальний курс</h1>
<div class='ml20'>
<div><span>Прелік навчальних курсів: </span>
<?php
    $Host = 'localhost';
    $User = 'root';
    $Password = '';
    $DBName = 'mysql';
    $Link = mysql_connect($Host, $User, $Password);
    mysql_select_db($DBName, $Link);
    // формування списку мов програмування
    isset($_GET['l'])? $t1 = $_GET['l']: $t1 = 1;
    $result = mysql_query('SELECT * FROM `languages`
                            WHERE 1');

    echo '<select id=lang name=lang>';
    while($row = mysql_fetch_array($result)){
        if($row['id'] == $t1){
            echo '<option
selected=selected>`.$row['name'].`</option>';
            value='.`.$row['id'].`
        }
    }
?>
```



```

}
$result = mysql_query('SELECT * FROM `states`
                        WHERE `states`.`id_prob`='.$tp');
echo '<select id=st name=st>';
if(mysql_affected_rows()<1) {
    echo '<option value=''>'.'-'</option>';
}
else{
    while($row = mysql_fetch_array($result)){
        echo '<option value=.`.$row['id'].`.`.$row['name'].`</option>';
    }
}
echo '</select>';
?><span id='changeS' class='change m110'>змінити</span></div><br/>
<div><span>Стан навчальної проблеми </span><input id='state' name='state'
type='text' size='100' autocomplete='off'/><span class='change'> <span id='editS'
class='edit' style='display:none;'>редагувати</span> <span id='delS' class='del'
style='display:none;'>видалити</span></div>
<input id='allstate' name='allstate' type='hidden' />
<table id='statelist'><tbody></tbody></table>
<div id='addBtnS' class='add'>+ Додати ще один стан</div><br/>
<br/><div><input id='resS' class='reset' type='button' value='Скинути' /><input
id='sendStat' type='button' value='Додати' /></div>
</div>
</form>
<form name='editLang' action='edit_lang.php' method='post'>
    <input id='eL_id' name='eL_id' type='hidden' value='' />
    <input id='eL_n' name='eL_n' type='hidden' value='' />
</form>
<form name='delLang' action='del_lang.php' method='post'>
    <input id='dL_id' name='dL_id' type='hidden' value='' />
</form>
<form name='editProb' action='edit_prob.php' method='post'>
    <input id='eP_l' name='eP_l' type='hidden' value='' />
    <input id='eP_id' name='eP_id' type='hidden' value='' />
    <input id='eP_n' name='eP_n' type='hidden' value='' />
</form>
<form name='delProb' action='del_prob.php' method='post'>
    <input id='dP_l' name='dP_l' type='hidden' value='' />
    <input id='dP_id' name='dP_id' type='hidden' value='' />
</form>
<form name='editState' action='edit_state.php' method='post'>
    <input id='eS_l' name='eS_l' type='hidden' value='' />
    <input id='eS_p' name='eS_p' type='hidden' value='' />
    <input id='eS_id' name='eS_id' type='hidden' value='' />
    <input id='eS_n' name='eS_n' type='hidden' value='' />
</form>
<form name='delState' action='del_state.php' method='post'>
    <input id='dS_l' name='dS_l' type='hidden' value='' />
    <input id='dS_p' name='dS_p' type='hidden' value='' />
    <input id='dS_id' name='dS_id' type='hidden' value='' />
</form>
<?php
    include 'bottom.php';
?>
</body>

<script type='text/javascript'>var g_forInitMas =
'course[k],corlist,allcor,problem[k],problast,allprob,state[k],statelist,allstate,s
endLang[m-inp],sendProb[m-inp],sendStat[m-
inp],lang[select],prob,st,addBtnL[m],addBtnP[m],addBtnS[m],resL[m],resP[m],resS[m],
changeL[m],editL[disp-m],delL[disp-m],changeP[m],editP[disp-m],delP[disp-
m],changeS[m],editS[disp-m],delS[disp-m]'</script>
<script type='text/javascript' src='js/DBD_c.js'></script>
<script type='text/javascript' src='js/lib.js'></script>
<script type='text/javascript' src='js/ajax.js'></script>
<script type='text/javascript' src='js/newCPS.js'></script>
</html>

```

statistics.php

```
<?php
```

```

    session_start();
    if(!isset($_SESSION['username']) || $_SESSION['right']<2){
        echo '<div>У вас немає прав доступу до сторінки або ви не <a
href='login.php'>увійшли в систему</a></div>';
        exit;
    }
?>
<html>
<head><title>Адміністративний модуль :: Статистика</title>
<meta http-equiv='Content-Type' content='text/html; charset=windows-1251' />
<link rel='stylesheet' type='text/css' href='css/main.css' />
<link rel='stylesheet' type='text/css' href='css/lib.css' />
<link rel='stylesheet' type='text/css' href='css/progressbar.css' />
<style>

</style>
</head>
<body>

<?php
    include 'logout.php';
?>
<div class='ml10'><a href='adminka.php'>повернутися</a></div><br/>
<h1>Статистика</h1>
<div id='instat' class='ml20'>
<div>Тип користувача
<select id='usertype' name='usertype'>
    <option value='1'>студент</option>
    <option value='2'>викладач</option>
    <option value='3'>розробник</option>
</select></div>
<div id='gS'>Група
<?php
    $Host = 'localhost';
    $User = 'root';
    $Password = '';
    $DBName = 'mysql';
    $Link = mysql_connect($Host, $User, $Password);
    mysql_select_db($DBName, $Link);
    $result = mysql_query('SELECT * FROM `groups`
                                WHERE 1');

    echo '<select id='group' name='group'>';
    echo '<option value=''>-</option>';
    while($row = mysql_fetch_array($result)){
        echo '<option value='.$row['id'].'>'.$row['name'].'</option>';
    }
    echo '</select>';
?></div>
<div>Перелік користувачів:
<select id='user' name='user' disabled='disabled'>
<option value=''>-</option>
</select></div>
<div>Загальна кількість сесій: <span id='Csession' class='b'>0</span></div>
<div>Cecii:
<select id='sess' name='sess' disabled='disabled'>
<option value=''>-</option>
</select></div>
<div>Загальна кількість пройдених навчальних курсів: <span id='Clang'
class='b'>0</span></div>
<div>Навчальні курси:
<select id='lang' name='lang' disabled='disabled'>
<option value=''>-</option>
</select></div>
<div>Загальна кількість порушених навчальних проблем: <span id='Cprob'
class='b'>0</span></div>
<div>Навчальні проблеми:
<select id='prob' name='prob' disabled='disabled'>
<option value=''>-</option>
</select></div>
<div>Загальна кількість порушених станів навчальних проблем: <span id='Cstate'
class='b'>0</span></div>

```

```

<div>Стани навчальних проблем:
<select id='state' name='state' disabled='disabled'>
<option value=''></option>
</select></div>
<div>Релевантні обмеження: <span id='Rconst' class='b'>0</span></div>
<div>Задоволені обмеження: <span id='Sconst' class='b'>0</span></div>
<div>Порушені обмеження: <span id='Vconst' class='b'>0</span></div>

</div>
<br/><hr/><br/>
<h1>Статистика по навчальних курсах</h1>
<div class='ml20 fs10'>Для активації списку слід вибрати користувача</div>
<div class='ml20 mb5'><span>Прелік навчальних курсів: </span>
<?php
    // формування списку мов програмування
    $result = mysql_query('SELECT * FROM `languages`
                           WHERE 1');
    echo '<select id='langS' name='langS' disabled='disabled'>';
    echo '<option value=''></option>';
    while($row = mysql_fetch_array($result)){
        echo '<option value='.$row['id'].'>'.$row['name'].'</option>';
    }
    echo '</select>';
?></div>
<div id='conAC' class='ml20' style='display:none;'>
<div class='mb5'>Релевантні обмеження: <span id='RconstA' class='b'>0</span></div>
<div class='mb5'>Задоволені обмеження: <span id='SconstA' class='b'>0</span></div>
<div>Порушені обмеження: <span id='VconstA' class='b'>0</span></div>
</div>
<?php
    include 'bottom.php';
?>

</body>

<script
    type='text/javascript'>var
    g_forInitMas
    =
    'usertype,group,gS[disp],user[inp],Csession,sess[inp],Clang,lang[inp],Cprob,prob[in
p],Cstate,state[inp],Vconst,Rconst,Sconst,langS[inp],conAC[disp],VconstA,RconstA,Sc
onstA'</script>
</html>

```