

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук  
(повна назва)

Кафедра Інформаційних управляючих систем  
(повна назва)

## КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти другий (магістерський)

Дослідження моделей і методів підвищення гнучкості розкладу ІТ-проєкту  
дистанційного навчання  
(тема)

Виконала:

студентка 2 курсу, групи УПГІТм-21-1  
Марина КУДРЯВЦЕВА  
(власне ім'я, прізвище)


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

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

Освітня програма Управління проєктами в  
галузі інформаційних технологій  
(повна назва освітньої програми)

Керівник проф. каф. ІУС, Максим ЄВЛАНОВ  
(посада, власне ім'я, прізвище)

Допускається до захисту  
Зав. кафедри


  
(підпис)

Костянтин ПЕТРОВ  
(власне ім'я, прізвище)

2023 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук  
Кафедра Інформаційних управляючих систем  
Рівень вищої освіти другий (магістерський)  
Спеціальність 122 Комп'ютерні науки  
(код і повна назва)  
Тип програми освітньо-наукова  
(освітньо-професійна або освітньо-наукова)  
Освітня програма Управління проектами в галузі інформаційних технологій  
(повна назва)

ЗАТВЕРДЖУІ   
Зав. кафедри \_\_\_\_\_  
(підпис)  
« 03 » квітня 2023 р.

**ЗАВДАННЯ**

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Кудрявцевій Марині Олександрівні  
(прізвище, ім'я, по батькові)

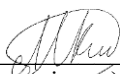
1. Тема роботи Дослідження моделей і методів підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання  
затверджена наказом університету від 03 квітня 2023 р. № 319Ст
2. Термін подання студентом роботи до екзаменаційної комісії 16 травня 2023 р.
3. Вихідні дані до роботи: науково-технічні публікації; джерела інтернету; науково-технічна література, що стосуються теми кваліфікаційної роботи; результати дослідження існуючого розкладу ІТ-проєкту компанії GreenForest.
4. Перелік питань, що потрібно опрацювати в роботі: аналіз існуючих моделей і методів підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання, розробка комбінованого методу та моделі підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання, дослідження особливостей реалізації комбінованого методу і моделі підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання, апробація комбінованого методу та моделі підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання.

## КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Аналіз існуючих моделей і методів підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання	03.04.2023	виконано
2	Обґрунтування мети створення комбінованого методу та постановка задачі	10.04.2023	виконано
3	Розробка комбінованого методу та моделі підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання	14.04.2023	виконано
4	Дослідження особливостей реалізації комбінованого методу і моделі підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання	17.04.2023	виконано
5	Апробація комбінованого методу та моделі підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання	22.04.2023	виконано
6	Оформлення пояснювальної записки та графічного матеріалу	28.04.2023	виконано
7	Підготовка презентації	04.05.2023	виконано
8	Подання студентом роботи для перевірки на плагіат	08.05.2023	виконано
9	Надання роботи на підпис науковому керівнику	10.05.2023	виконано
10	Попередній захист роботи	11.05.2023	виконано
11	Надання роботи на рецензію	12.05.2023	виконано
12	Надання роботи на підпис завідувачу кафедри	15.05.2023	виконано
13	Захист кваліфікаційної роботи	17.05.2023	виконано

Дата видачі завдання 03 квітня 2023 р.

Студент \_\_\_\_\_

  
(підпис)

Керівник роботи \_\_\_\_\_

  
(підпис)

проф. каф. ІУС Максим ЄВЛАНОВ  
(посада, власне ім'я, прізвище)

## РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи: 135 стор., 40 рис., 24 табл., 30 джерел, 1 додаток.

АЛГОРИТМ НА ОСНОВІ ОБМЕЖЕНЬ, ГЕНЕТИЧНИЙ АЛГОРИТМ, ГНУЧКИЙ РОЗКЛАД ЗАНЯТЬ, ДИСТАНЦІЙНЕ НАВЧАННЯ, ЖАДІБНИЙ АЛГОРИТМ, ЗАВДАННЯ ВИКОНАННЯ ОБМЕЖЕНЬ, ІТ-ПРОЄКТ, КОМБІНОВАНИЙ МЕТОД ПІДВИЩЕННЯ ГНУЧКОСТІ РОЗКЛАДУ, ФІКСОВАНИЙ РОЗКЛАД ЗАНЯТЬ, ШТУЧНА НЕЙРОННА МЕРЕЖА.

Метою даної роботи є дослідження моделей і методів підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання та створення комбінованого методу і моделі, які мають використовувати сильні сторони існуючих, мінімізуючи при цьому їхні недоліки.

Об'єктом дослідження в рамках магістерської кваліфікаційної роботи є процес формування розкладу ІТ-проєкту дистанційного навчання.

Предметом дослідження являються моделі та методи аналізу вирішення задачі розкладу.

Теоретичними результатами дослідження є опис комбінованого методу та його основних етапів.

Практичним результатом є апробація комбінованого методу і моделі підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання, на прикладі фірми дистанційних курсів.

Новизна дослідження полягає в дослідженні та розробці комбінованого методу і моделі, сценаріїв їхнього використання, описі взаємодії користувачів з ними, описі їхньої реалізації на платформно-незалежному рівні.

## ABSTRACT

The explanatory note to the qualification work: 135 p., 40 fig., 24 tabl., 30 sources, 1 appendix.

ARTIFICIAL NEURAL NETWORK, COMBINED SCHEDULING FLEXIBILITY METHOD, CONSTRAINT SATISFACTION PROBLEM, CONSTRAINT-BASED ALGORITHM, DISTANCE LEARNING, FIXED SCHEDULING, FLEXIBLE SCHEDULING, GENETIC ALGORITHM, GREEDY ALGORITHM, IT-PROJECT.

The purpose of this work is to investigate models and methods for increasing the flexibility of the IT project schedule of distance learning and to create a combined method and model that should use the strengths of the existing ones, while minimizing their shortcomings.

The object of research within the framework of the master's qualification work is the process of forming the schedule of the distance learning IT project.

The subject of the research is the methods and models of analysis and solution of the scheduling problem.

The theoretical results of the research are a description of the combined method and its main stages.

The practical results are the approbation of the combined method and the model of increasing the flexibility of the distance learning IT project schedule, on the example of the distance course company.

The novelty of the research is the research and development of the combined method and model, scenarios of their use, description of user interaction with them, description of their implementation at the platform-independent level.

## ЗМІСТ

Скорочення та умовні позначки	9
Вступ	10
1 Аналіз існуючих моделей і методів підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання	12
1.1 Аналіз актуальності дослідження моделей і методів підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання	12
1.2 Аналіз існуючих рішень підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання	15
1.3 Аналіз існуючих моделей та методів підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання	22
1.4 Обґрунтування мети створення комбінованого методу та моделі підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання, формування вимог до них	29
1.5 Постановка задачі дослідження моделей і методів підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання	37
2 Розробка комбінованого методу та моделі підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання	38
2.1 Опис комбінованого методу підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання, етапи створення розкладу	38
2.2 Використання штучної нейронної мережі для розробки комбінованого методу підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання	41
2.3 Використання алгоритму на основі обмежень для розробки комбінованого методу підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання	47

2.4 Використання генетичного алгоритму для розробки комбінованого методу підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання	54
2.5 Використання жадібного алгоритму для розробки комбінованого методу підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання	60
2.6 Висновки з розробки комбінованого методу підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання	63
3 Дослідження особливостей реалізації комбінованого методу і моделі підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання	65
3.1 Сценарій використання комбінованого методу і моделі підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання	65
3.2 Опис взаємодії користувачів з комбінованим методом за допомогою User-story	71
3.3 Опис реалізації комбінованого методу на платформно-незалежному рівні	77
3.4 Висновки з дослідження особливостей реалізації комбінованого методу підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання	81
4 Апробація комбінованого методу та моделі підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання	83
4.1 Аналіз предметної галузі, що визначається діяльністю фірми дистанційних курсів	83
4.2 Визначення сфери застосування комбінованого методу підвищення гнучкості розкладу ІТ-проєкту для фірми дистанційних курсів	86
4.3 Аналіз використання комбінованого методу підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання у порівнянні з фіксованим графіком	87

4.3.1	Вихідні дані для комбінованого методу, що пов'язані з діяльністю фірми дистанційних курсів	87
4.3.2	Етап комбінованого методу «1. Формування обмежень»	92
4.3.3	Етап комбінованого методу «2. Підготовка вхідних даних на основі вводу користувача»	93
4.3.4	Етап комбінованого методу «3. Формування розкладу», під-етап «3.1. Пріоритизація занять»	98
4.3.5	Етап комбінованого методу «3. Формування розкладу», під-етап «3.2. Обробка конфліктів»	99
4.4	Висновки з четвертого розділу кваліфікаційної роботи	103
	Висновки	104
	Перелік джерел посилання	106
	Додаток А «Графічний матеріал до магістерської роботи»	108

## СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

- АІП – аналітичний ієрархічний процес;
- ГРЗ – гнучкий розклад занять;
- ДН – дистанційне навчання;
- ЗВО – завдання виконання обмежень;
- ФРЗ – фіксований розклад занять;
- ШНМ – штучні нейронні мережі;
- АС – Acceptance Criteria;
- DAL – Data Access Layer;
- FIFO – first in first out;
- INVEST – independent, negotiable, valuable, estimable, small, and testable;
- PMBOK – Project Management Body of Knowledge’s;
- ReLU – Rectified Linear Unit;
- SOLID – S: Single Responsibility Principle, O: Open/Closed Principle, L: Liskov Substitution Principle, I: Interface Segregation Principle, D: Dependency Inversion Principle;
- UML – Unified Modeling Language.

## ВСТУП

Сьогодні розвиток інформаційних технологій дозволяє вирішувати актуальні проблеми модернізації систем забезпечення навчання. Суть такої модернізації перш за все стосується концепції дистанційної навчання (ДН), яка, завдяки використанню глобальної мережі Інтернет, дає широкий доступ до ресурсів освіти. Особливого значення така модернізація систем забезпечення ДН набуває в Україні у період пандемії та воєнного стану.

Починаючи від процесів планування і закінчуючи контролем навчання, на фірмі дистанційних курсів доцільно використовувати автоматизовані технології. Ця необхідність в автоматизації має на меті підвищення швидкості і якості роботи з контингентом студентів та вчителів, забезпеченням надійності і цілісності управління в цілому, обумовлює розгляд фірми дистанційних курсів у якості ІТ-проєкту.

На сьогоднішній день існує багато різноманітних дистанційних курсів. Але майже усі вони платні або пропонують лише один напрямок навчання. Процес навчання в таких системах зазвичай проходить за двома сценаріями. Перший з них має на увазі що усі матеріали викладені одразу. Для комунікації та запитань існує чат з іншими студентами курсу. Другий вид підпорядковується жорсткому графіку, заздалегідь встановленим викладачем або адміном.

Як наслідок проведення занять за таким жорстким графіком можна виділити: нерівномірне навантаження викладачів, створення не оптимальної кількості занять, часткову відвідуваність та з цього ж неповне засвоєння матеріалу студентами, нестачу викладачів, зацікавлених у співпраці за сумісництвом та інші подібні супутні проблеми.

Дистанційні курси в наш час мають надавати послуги студентам з підтриманням саме гнучкого розкладу занять. При цьому повинен зберігатися зв'язок з викладачем, так зване «живе спілкування», задля

найкращого засвоєння матеріалу. Тобто існує потреба в створенні гнучкого розкладу занять. Через велику кількість змінних виникає потреба в автоматизації.

Отже, складання гнучкого розкладу занять для невеликих груп студентів спрямовано на підтримання переваг дистанційного навчання, коли студент має змогу навчатись в зручний для нього час, не втрачаючи контакту та підтримки з боку викладача. Викладач, в свою чергу, зберігає за собою гнучкий графік віддаленої праці з усіма її перевагами.

Об'єктом дослідження в рамках магістерської кваліфікаційної роботи є процес формування розкладу ІТ-проєкту дистанційного навчання.

Предметом дослідження являються моделі та методи аналізу вирішення задачі розкладу.

Метою даної роботи є дослідження моделей і методів підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання та створення комбінованого методу і моделі, які мають використовувати сильні сторони існуючих, мінімізуючи при цьому їхні недоліки.

Магістерська кваліфікаційної робота виконана згідно з методичними вказівками 2019 року щодо розробки та оформлення магістерської кваліфікаційної роботи за спеціальністю 122 Комп'ютерні науки та освітньою програмою «Управління проєктами в галузі інформаційних технологій» [1].

# 1 АНАЛІЗ ІСНУЮЧИХ МОДЕЛЕЙ І МЕТОДІВ ПІДВИЩЕННЯ ГНУЧКОСТІ РОЗКЛАДУ ІТ-ПРОЄКТУ ДИСТАНЦІЙНОГО НАВЧАННЯ

1.1 Аналіз актуальності дослідження моделей і методів підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання

Сьогодні розвиток інформаційних технологій дозволяє вирішувати актуальні проблеми модернізації систем забезпечення навчання. Суть такої модернізації перш за все стосується концепції дистанційної навчання (ДН), яка, завдяки використанню глобальної мережі Інтернет, дає широкий доступ до ресурсів освіти. Особливого значення така модернізація систем забезпечення ДН набуває в Україні у період пандемії та воєнного стану [2].

Починаючи від процесів планування і закінчуючи контролем навчання, на фірмі дистанційних курсів доцільно використовувати автоматизовані технології. Ця необхідність в автоматизації має на меті підвищення швидкості і якості роботи з контингентом студентів та вчителів, забезпеченням надійності і цілісності управління в цілому, обумовлює розгляд фірми дистанційних курсів у якості ІТ-проєкту [2].

На сьогоднішній день існує багато різноманітних дистанційних курсів. Але майже усі вони платні або пропонують лише один напрямок навчання. Процес навчання в таких системах зазвичай проходить за двома сценаріями. Перший з них має на увазі що усі матеріали викладені одразу. Для комунікації та запитань існує чат з іншими студентами курсу. Другий вид підпорядковується жорсткому графіку, заздалегідь встановленим викладачем або адміністратором [2].

Дистанційні курси в наш час мають надавати послуги студентам з підтриманням саме гнучкого розкладу занять. При цьому повинен зберігатися зв'язок з викладачем, так зване «живе спілкування», задля найкращого засвоєння матеріалу. Тобто існує потреба в створенні гнучкого

розкладу занять (ГРЗ). Через велику кількість змінних виникає потреба в автоматизації [2].

Отже, складання ГРЗ для невеликих груп студентів спрямовано на підтримання переваг дистанційного навчання, коли студент має змогу навчатись в зручний для нього час, не втрачаючи контакту та підтримки з боку викладача. Викладач, в свою чергу, зберігає за собою гнучкий графік віддаленої праці з усіма її перевагами [2].

Завдання створення ГРЗ передбачає організацію набору завдань або подій протягом певного періоду часу з метою максимізації продуктивності та ефективності з урахуванням будь-яких відповідних обмежень. Це включає:

- визначення завдань або подій, які потрібно запланувати;
- визначення їх тривалості та черговості;
- оцінка часу, необхідного для виконання кожного завдання;
- визначення будь-яких залежностей або передумов між завданнями;
- розподіл ресурсів, таких як персонал або обладнання, для кожного завдання;
- облік будь-яких обмежень у розкладі, таких як доступність персоналу чи ресурсів, або обмеження щодо часу виконання певних завдань;
- створення шкали часу або календаря, який візуально представляє розклад і дозволяє легко відстежувати та контролювати прогрес.

Конкретні деталі процесу створення розкладу можуть відрізнятися залежно від характеру запланованих завдань або подій, а також від конкретних інструментів або методів, які використовуються для створення розкладу. Однак загальна мета полягає в тому, щоб створити план, який дозволить ефективно та результативно виконати завдання або організувати події в межах доступного часу.

Для аналізу актуальності дослідження моделей і методів підвищення гнучкості розкладу IT-проєкту дистанційного навчання було використано

матрицю SWOT-аналізу.

SWOT – один із найпоширеніших аналітичних методів, який дозволяє в комплексі оцінити сильні й слабкі сторони фірми, а також можливості й загрози, що впливають на неї [3].

SWOT-аналіз процесу підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання представлений на рис. 1.1.

<b>Сильні сторони (Strengths)</b>	<b>Можливості (Opportunities)</b>
Гнучкий графік для студентів	Підвищити прибуток
Гнучкий графік для викладачів	Збільшити аудиторію
Ефективне розподілення занять курсу	Доцільно розподілити навантаження викладачів
<b>Слабкі сторони (Weaknesses)</b>	<b>Загрози (Threats)</b>
Необхідність вносити додаткову інформацію	Брак бюджету на створення та впровадження ГРЗ
Необхідність адаптуватися під новий підхід адміністраторам та користувачам	

Рисунок 1.1 – SWOT-аналіз підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання

Отже, впровадження ГРЗ може підвищити прибуток та збільшити аудиторію студентів за рахунок того, що більша кількість людей зможе відвідувати курси. Крім того, може підвищитися рівень конкурентоспроможності, студенти будуть в першу чергу дивитися на курси з гнучким розкладом. Також за рахунок кращого розподілення занять, фірма зможе вести більше курсів одночасно. Це, в свою чергу, також має підвищити прибуток. ГРЗ доповнення до існуючих систем, тобто впровадження його не вимагає призупинення праці.

## 1.2 Аналіз існуючих рішень підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання

Наразі через пандемію коронавірусу та воєнний стан в Україні все більшої популярності набувають курси, що мають крім аудиторних занять ще й дистанційний аналог.

Найпопулярнішими з напрямлень таких курсів є курси іноземних мов, прикладом таких проєктів є ІТ проєкт дистанційних курсів української компанії GreenForest рис 1.2 [4].

**GREEN FOREST**

ДИСТАНЦІЙНИЙ КУРС \* MY GREEN FOREST КОНТ

4 МІСЯЦІ  
СТАРТ: 6.02 | 6.03

## ДИСТАНЦІЙНИЙ КУРС СТАНДАРТ

3 зан./тиж.\*1 год. 30 хв.  
2 Communication Classes: Пн./Ср. або Вт./Чт.  
1 Grammar Class: Пт. або Сб.  
Графік: 8:10 / 9:10 / 9:50 / 10:50 / 15:00 /  
16:40 / 18:20 / 20:00

---

- ДИСТАНЦІЙНО:** онлайн курс англійської можна пройти з будь-якої точки України та світу, де є Інтернет
- ЕФЕКТИВНО:** навчання в групах в атмосфері реального заняття у класі
- КОМФОРТНО:** цілий рівень за 4 місяці і 50 занять
- ВИГІДНО:** 92 грн. за 1 год. навчання + економія коштів та часу на дорогу
- ІННОВАЦІЙНО:** навчання у програмі Zoom на базі гібридної інтерактивної платформи Notes by Green Forest + контроль за навчанням у додатку My Green Forest
- ЗРУЧНО:** навчатись можна не виходячи з дому, роботи чи улюбленого кафе
- ПРОФЕСІЙНО:** всі викладачі Green Forest проходять підготовку в Green Forest Teaching Centre by Nadia Diachuk
- БЕЗКОШТОВНІ БОНУСИ:** доступ до онлайн версії підручника, онлайн лекції на різні теми та інші англомовні івенти щотижня

Рисунок 1.2 – Приклад дистанційного курсу від GreenForest

Гнучкість розкладу тут полягає у обранні зручного часу з запропонованого, також є вибір днів тижня. Недоліком такого підходу є прив'язка до певної схеми днів та необхідність вчителям вести заняття навіть в маленьких групах, маючи перевантаженими інші робочі часи. З таким підходом вчителі повинні бути на зв'язку кожний запропонований часовий інтервал занять оптимально це чи ні.

Також до інструментів відображення та створення розкладу можна віднести поштовий сервіс Google – Gmail, а також Outlook, який входить до складу пакету Microsoft 365 [5, 6].

Коли ви створюєте нову зустріч, Gmail має функцію «Запропоновані години» рис. 1.3, яка створює список проміжків, коли ні у вас, ні в інших учасників немає інших подій у поточний період.

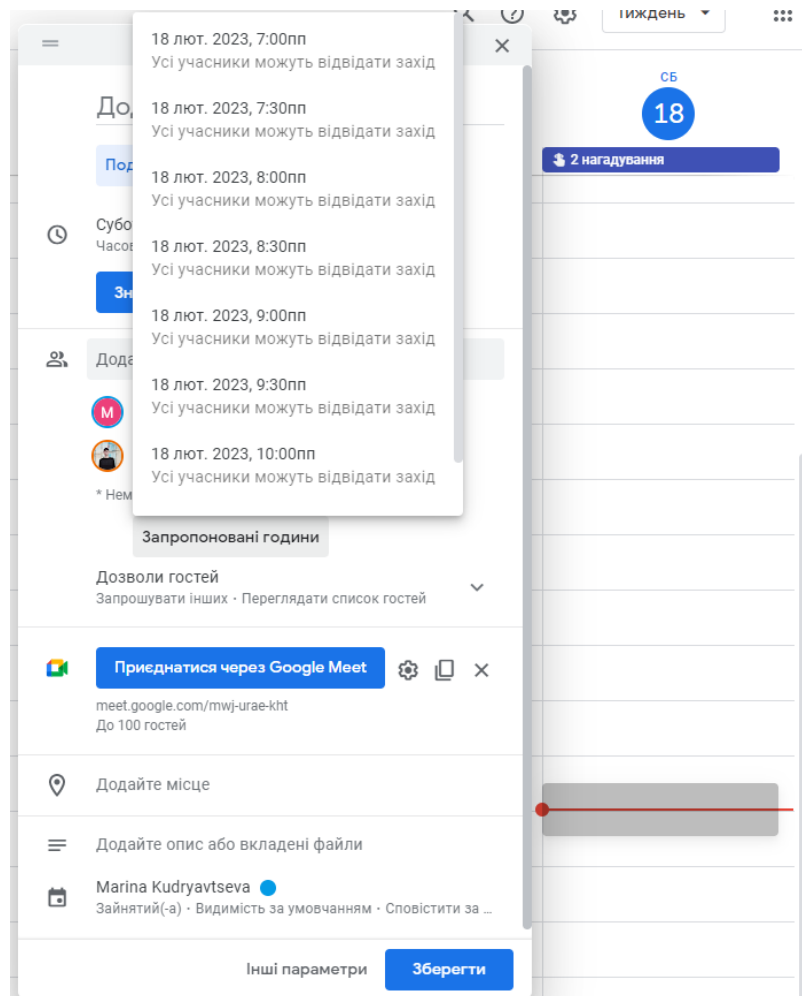


Рисунок 1.3 – Функція Gmail «Запропоновані години»

Також, Gmail має функцію відображення подій конкретного чи декількох учасників зустрічі для підбору вдалого часу власноруч рис 1.4.

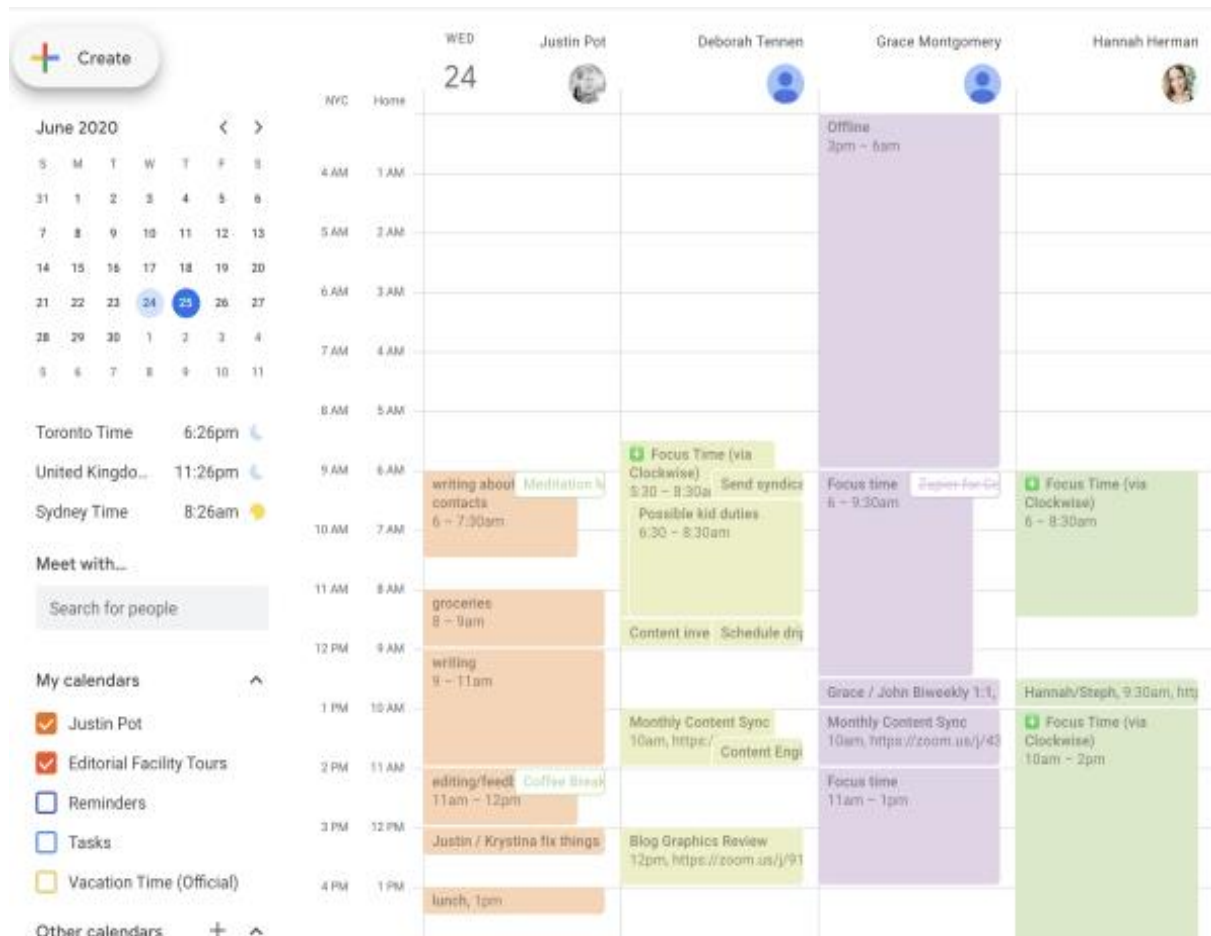


Рисунок 1.4 – Режим Gmail для ручного вибору часу зустрічей

У такому режимі події кожного відображаються один проти одного різними кольорами, відображення стислої інформації дає нагоду вирішувати, що можна сумістити чи здвинути у розкладі [5].

На відміну від Gmail, Outlook не буде генерувати список на вибір і просто створить подію у першому вільному місці. Потім час можливо скорегувати й перенести в ручному режимі [6].

Outlook окрім переліченого має градацію учасників зустрічі на «Обов'язкових» та «Необов'язкових», що враховується при авто підборі часу у вигляді відповідних пунктів «Обов'язкові учасники», «Обов'язкові учасники і ресурс (кімната)», «Всі учасники і ресурс» та «Всі учасники та усі

ресурси». Під ресурсом можуть виступати кімнати чи обладнання, їх доступність відслідковується автоматично рис 1.5 [6].

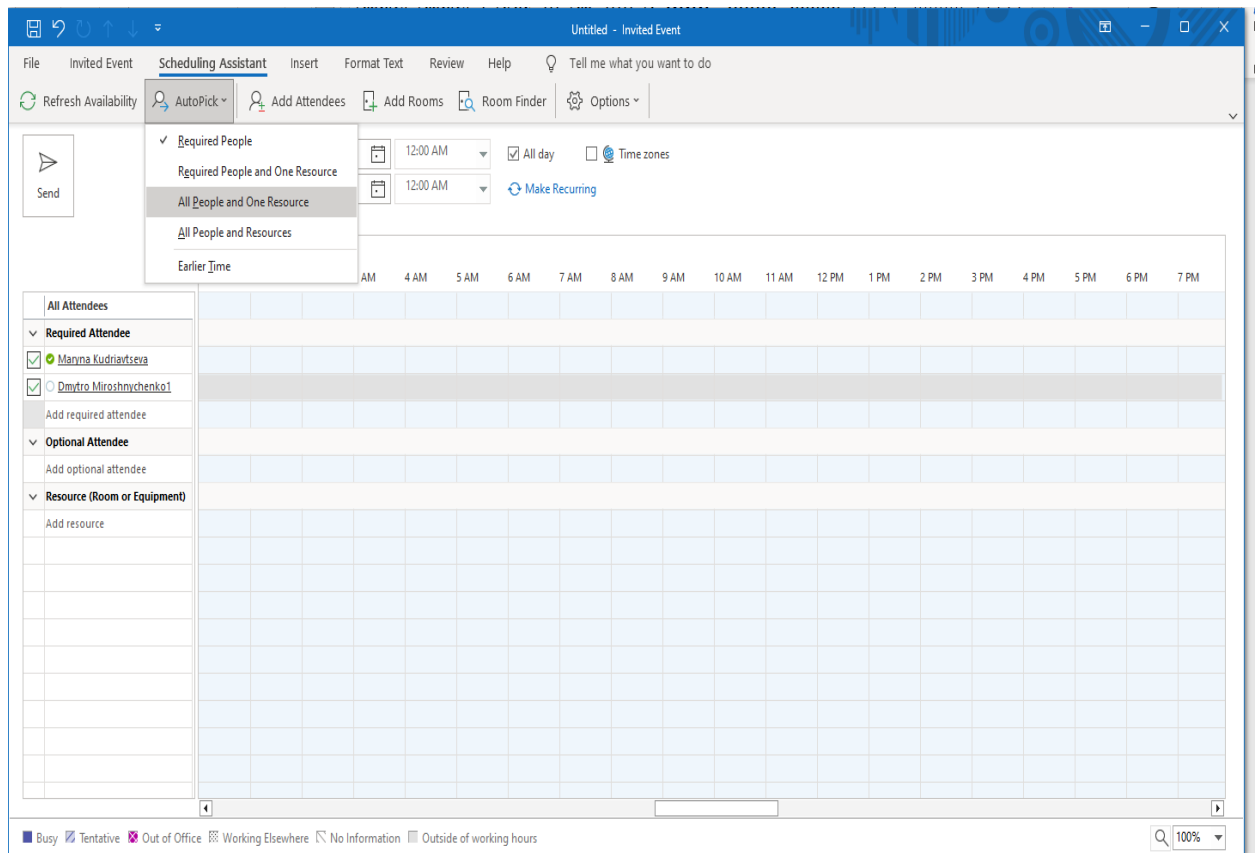


Рисунок 1.5 – Режим автоматичного підбору часу зустрічі в Outlook

Додаток ascTimeTables покликаний автоматизувати створення розкладу для великих навчальних закладів таких як школи, гімназії, ліцеї, коледжі, ВУЗи або інших навчальних закладів протягом усього навчального року [7].

ascTimeTables – це десктопний додаток, ліцензія на нього помісячна. Існують як обмежені так і повні версії додатку в залежності від ціни підписки. Ліцензія йде відразу на декілька закладів в залежності від типу, початкові школи, середні, коледжі тощо [7].

Додаток дає можливість створити розклад чи навчальну програму для аудиторій та викладачів, беручи до уваги тижневі обмеження часу, призначені кожному з викладачів. Програма передбачає всі типи спеціальних завдань, таких як додавання кількох викладачів до одного предмета.

Програма самостійно конфігурує аудиторії та викладачів, підказує чи є можливість поставити певний предмет на конкретну годину або, наприклад, чи вільний викладач на певну годину рис 1.6 [7].



Рисунок 1.6 – Сайт ascTimeTables з прикладом головного екрану програми

Кожне з перелічених рішень має свої власні недоліки.

GreenForest – не раціональне використання часу викладачів, що виливається у відсутність гнучкості для інших типів дистанційних курсів рис 1.7.

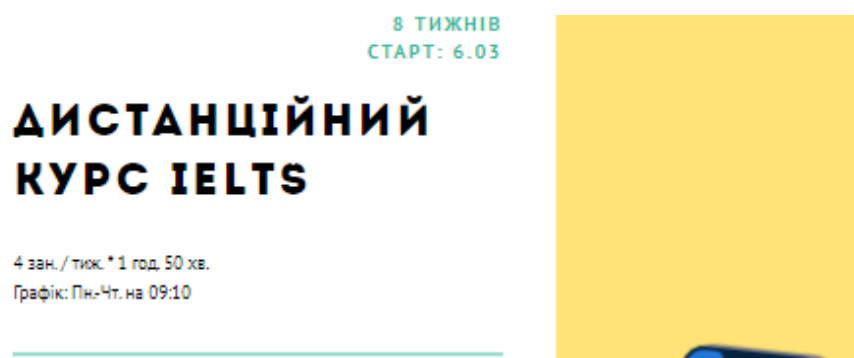


Рисунок 1.7 – Приклад курсу GreenForest без гнучкого розкладу занять

Gmail та Outlook допомагають створити розклад, але не є повноцінними інструментами для створення графіку, так як не мають обмежень по навантаженню, не обирають оптимальні відрізки часу для групи, тощо.

Недоліком ascTimeTables є створення розкладів на семестри/курси. А також, відсутність взаємодії з розкладами учнів, що пов'язані з їх повсякденної активністю. Цей недолік робить додаток ascTimeTables не зручним для використання фірмами дистанційних курсів в якості рішення для створення ГРЗ.

Результати порівняння рішень наведено у вигляді таблиці 1.1.

Таблиця 1.1 – Порівняння існуючих рішень підвищення гнучкості розкладу проекту дистанційного навчання

Критерії порівняння рішень	GreenForest	Gmail (Google Calendar)	Outlook	ascTimeTables
побажання студентів щодо розкладу	враховуються	-	-	тільки у преміум версії,
тип рішення	ІТ-проект	інструмент	інструмент	десктопний додаток
побажання викладачів щодо розкладу	не враховуються	-	-	враховуються, але загальні. Як то не працювати по понеділках чи у першій половині дня, тощо.
створення розкладу автоматизовано	ні, розклад зафіксований по днях. Гнучкість полягає в наявності варіантів днів та часу.	є функціонал підказок інтервалів для полегшення обирання часу занять в ручному режимі	є функціонал авто розміщення 1-го заняття, але його все ще потрібно корегувати в ручному режимі	так

Кінець таблиці 1.1

Критерії порівняння рішень	GreenForest	Gmail (Google Calendar)	Outlook	ascTimeTables
спосіб переглянути вільні проміжки часу усіх учасників заняття	ні	так	так	ні
час занять розподілений доцільно	не доцільно, викладач може проводити заняття для малої аудиторії в один час і мати переповненими інші часи. Тобто один курс буде гнучким, а для інших курсів не вистачить викладачів для забезпечення цього підходу.	не має такого функціоналу	не має такого функціоналу	доцільно, але без урахування вільного часу студентів
спосіб автоматично знайти вільний час з урахуванням зайнятості усіх учасників заняття	ні	так	так	ні

Отже, комбінований метод та модель підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання повинні зберегти плюси наведених аналогів, а також виправити їх недоліки, щоб досягти мети – доцільно розподіляти заняття курсу враховуючи побажання усіх їх учасників та наявні обмеження в автоматичному режимі.

### 1.3 Аналіз існуючих моделей та методів підвищення гнучкості розкладу ІТ-проекту дистанційного навчання

Теорія розкладів має велику кількість алгоритмів-методів розв'язання задачі створення розкладу, в тому числі і гнучкого його варіанту.

Жадібний алгоритм (з англ. *eager*) є простим, але ефективним підходом для створення гнучкого графіка. Алгоритм працює, призначаючи завдання доступним часовим інтервалам за принципом FIFO (*first in first out*) «першим прийшов, першим пішов», коли завдання плануються якнайшвидше, не залишаючи жодних проміжків. Ось огляд високого рівня того, як алгоритм *eager* можна застосувати в контексті створення гнучкого графіка у розрізі етапів його реалізації [8]:

- визначення часових інтервалів, доступних для планування: у випадку курсу дистанційного навчання це може передбачати збір інформації від студентів про їхню доступність, а потім створення списку доступних часових інтервалів;

- збір списку завдань, які потрібно запланувати: у контексті курсу дистанційного навчання це може включати збір списку лекцій, завдань та інших заходів, які потрібно виконати;

- пріоритизація завдань: починаючи із завдання з найвищим пріоритетом розставляти завдання на перший доступний проміжок часу, який є достатньо довгим для виконання завдання, процес повторюється, доки не буде заплановано всі завдання;

- обробка конфліктів: якщо є конфлікти, через які два чи більше завдань не можна запланувати в один часовий проміжок, визначте пріоритети завдань на основі їх важливості та кінцевих термінів і, якщо необхідно, перенесіть їх на інші часові проміжки;

— оптимізація розкладу: коли всі завдання заплановано, розклад можна оптимізувати, щоб мінімізувати прогалини або максимально використати доступні часові інтервали здвинувши заняття в ту чи іншу сторону, змінивши пріоритети.

Жадібний алгоритм є відносно простим підходом до створення гнучкого розкладу, але він може бути ефективним для вирішення багатьох проблем планування, особливо коли кількість завдань є відносно невеликою та існує небагато обмежень щодо планування. Однак складніші проблеми планування можуть вимагати більш складних алгоритмів, які можуть обробляти складніші обмеження та цілі оптимізації [8].

Алгоритм на основі обмежень — це ще один підхід до створення гнучкого графіка, який може бути корисним у випадках, коли існує багато обмежень, які потрібно враховувати. Ось огляд високого рівня того, як алгоритм на основі обмежень можна застосувати в контексті створення гнучкого графіка [9]:

— визначення обмежень, які необхідно враховувати під час створення розкладу, у контексті курсу дистанційного навчання це може включати такі обмеження, як наявність студентів і викладачів, кількість доступних часових проміжків і будь-які обмеження щодо тривалості лекцій або завдань;

— створення набору правил, які керуватимуть плануванням завдань, наприклад, що лекції можуть бути заплановані лише протягом певних проміжків часу або що завдання мають бути виконані до певних термінів;

— створення набору можливих розкладів на основі обмежень і правил, що може передбачати створення списку всіх можливих комбінацій завдань і часових інтервалів, які задовольняють обмеженням і правилам;

— оцінка кожного розкладу на основі набору критеріїв ефективності;

— обрання остаточного розкладу, тобто розклад із найвищим балом за критеріями ефективності, при цьому якщо існує кілька розкладів з однаковою оцінкою, можна вибрати той, який задовольняє додаткові критерії;

— оптимізація розкладу.

Алгоритм на основі обмежень може бути корисним у випадках, коли існує багато обмежень, які необхідно враховувати під час створення розкладу. Однак його реалізація може бути складнішою і не завжди може призвести до найоптимальнішого розкладу [9].

Генетичний алгоритм – ще один підхід, який можна використовувати для створення гнучкого графіка. Ось огляд високого рівня того, як генетичний алгоритм можна застосувати в контексті створення гнучкого графіка [10]:

— визначення проблеми, яка в даному випадку полягає у створенні гнучкого графіка, який відповідає певним критеріям: критерії можуть включати такі речі, як забезпечення того, щоб усі заняття були заплановані протягом певного періоду часу, або щоб студенти були рівномірно розподілені між різними курсами;

— створення початкової популяції: після визначення проблеми можна створити початкову популяцію можливих розкладів, при цьому кожен розклад має представляти рішення проблеми;

— визначення функції відповідності: функція відповідності використовується для оцінки того, наскільки добре кожна хромосома (або графік) вирішує проблему (функція повинна враховувати критерії, які були визначені на кроці 1, наприклад часові рамки, протягом яких мають бути заплановані заняття, або рівномірний розподіл студентів);

— застосування генетичних операторів: генетичні оператори (кросовер і мутація) використовуються для створення нових хромосом із існуючої популяції;

— оцінка нової популяції: функція відповідності застосовується до нової популяції, щоб оцінити, наскільки добре кожна хромосома вирішує проблему;

— повторення двох попередніх кроків: доки не буде виконано критерій зупинки, тобто може бути певна кількість поколінь або певний рівень придатності для найкращої хромосоми в популяції;

— обрання найкращого рішення: після завершення роботи генетичного алгоритму найкраще рішення (або хромосому) можна вибрати як остаточний розклад.

Генетичний алгоритм може бути корисним у випадках, коли існує багато можливих рішень проблеми, і коли може бути важко знайти оптимальне рішення за допомогою традиційних методів. Однак генетичні алгоритми можуть бути дорогими в обчислювальному плані та вимагати великої кількості ітерацій, щоб знайти найкраще рішення. Крім того, функція відповідності має бути ретельно розроблена, щоб забезпечити точне відображення проблеми, що вирішується [10].

Штучні нейронні мережі (ШНМ) можна використовувати в задачі планування для прогнозування та оптимізації розкладів на основі історичних даних і вхідних даних у реальному часі.

ШНМ – це обчислювальні моделі, розроблені для імітації функціонування людського мозку, і їх можна навчити розпізнавати шаблони в даних і робити прогнози на основі цих шаблонів [11].

У контексті планування ШНМ можна навчити на історичних даних, щоб вивчити шаблони в процесі планування, такі як оптимальне планування завдань і розподіл ресурсів.

Загалом, ШНМ можуть бути потужним інструментом для завдання планування, що дозволяє точніше та ефективніше планувати на основі історичних даних і даних у реальному часі. Тобто в задачі складання розкладу вони можуть бути використанні для вирішення конфліктів даних.

Аналіз використання існуючих моделей та методів для створення комбінованого методу підвищення гнучкості розкладу ІТ-проекту дистанційного навчання представлено у таблиці 1.2.

Таблиця 1.2 – Аналіз використання моделей та методів за етапами складання ГРЗ

Основні етапи створення ГРЗ		Моделі та методи	Описання проблеми	Результати
1. Формування обмежень		Алгоритм на основі обмежень, ШНМ	Необхідно визначити всі обмеження та на їх основі вивести критерії ефективності та загальні умови вибору та пріоритизації.	Обмеження, умови, критерії ефективності
2. Підготовка вхідних даних		Генетичний алгоритм	Дані можуть містити повтори інформації (наприклад, відрізок часу з 1-5, та з 2-3 => 1-5). Також інформація може бути не вірною чи не достатньої для створення розкладу.	Вхідні дані отриманні за допомогою генетичних операторів кросоверу та мутації
3. Формування розкладу	Пріоритизація завдань	Жадібний алгоритм	Спираючись на критерії ефективності отриманих на етапі 1 треба провести пріоритизацію для реалізації так званого «жадібного вибору»	Варіант ГРЗ
	Обробка конфліктів	ШНМ	У разі неповних чи помилкових даних розклад може бути неможливо створити	Доповнена чи/або змінена модель ГРЗ

Підводячи підсумок, різні алгоритми мають різні сильні та слабкі сторони, і деякі можуть бути більш придатними для певних типів проблем планування, ніж інші. Наприклад, алгоритм на основі обмежень може бути корисним для планування завдань зі складними обмеженнями, тоді як генетичний алгоритм може бути більш ефективним для планування проблем, які включають велику кількість змінних.

Крім використовуваного алгоритму, на ефективність алгоритму планування також можуть впливати такі фактори, як розмір проблеми планування, кількість обмежень і кількість доступних ресурсів.

Отже, комбінований метод буде використовувати сильні сторони розглянутих алгоритмів та моделі:

- простий підхід, що легко автоматизується;
- вміння працювати з великою кількістю обмежень;
- вміння опрацювати велику кількість вхідних даних та їх варіантів;
- завдання плануються якнайшвидше, не залишаючи жодних проміжків;
- здатність доповнювати дані, за потреби, наближеними до реальних.

Недоліки існуючих моделей та методів підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання:

- кількість завдань повинна бути невеликою;
- може опрацювати лише невелику кількість обмежень;
- більшість методів складна в реалізації та повільна (тобто, повинно пройти дуже багато ітерацій, щоб отримати кінцевий результат).

При цьому об'єднання розглянутих існуючих моделей та методів підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання компенсує їхні недоліки й робить метод більш універсальним, тобто здатним вирішувати різнотипні задачі складання ГРЗ.

Виходячи з вище переліченого, сильні сторони кожного методу будуть використанні на різних кроках реалізації жадібного алгоритму, дані і умови для якого будуть отримані за допомогою алгоритму на основі обмежень та генетичного алгоритму. Проблему конфліктних та недостатніх даних вирішить ШНМ рис. 1.8.

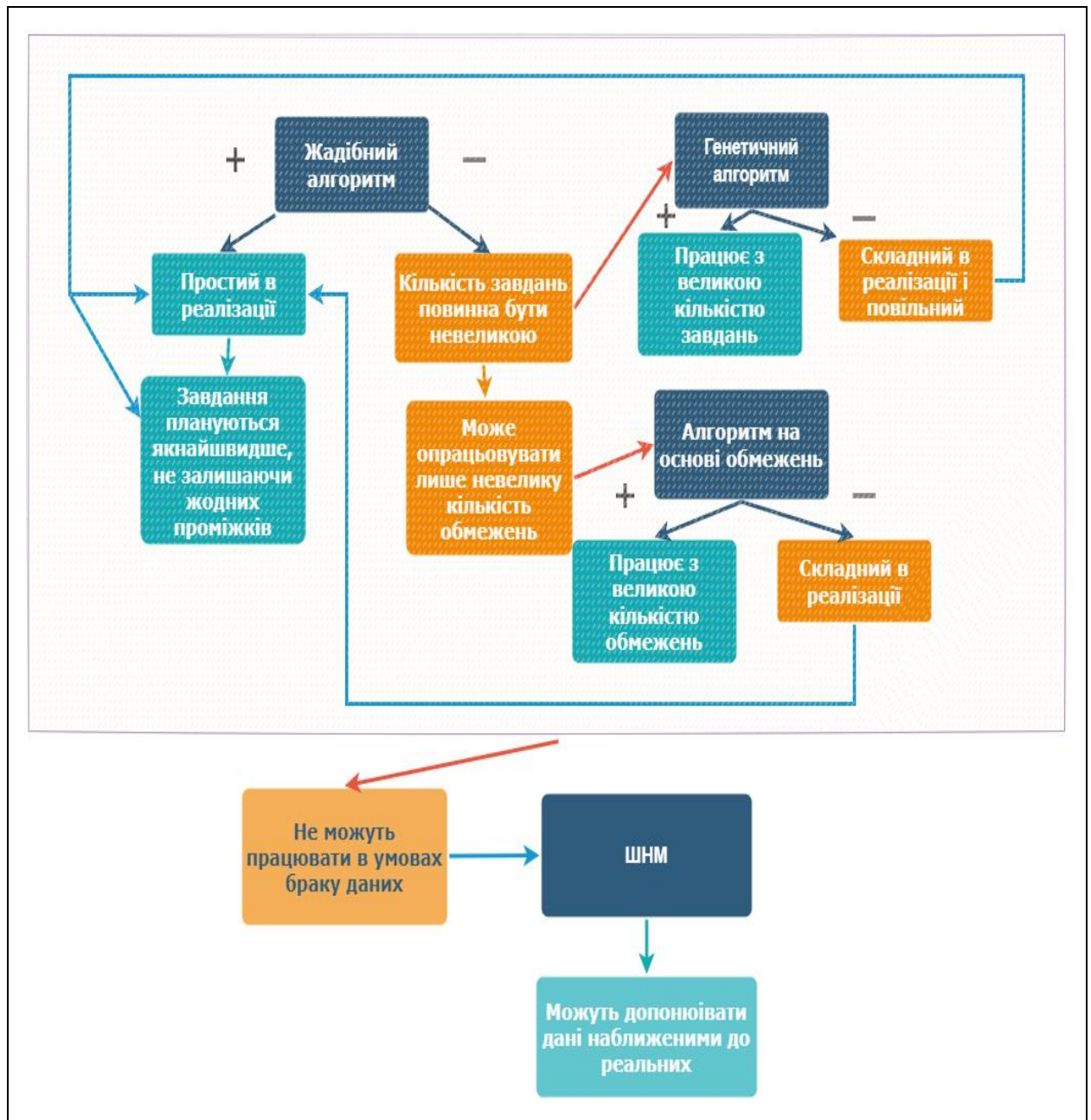


Рисунок 1.8 – Схема компенсування недоліків існуючих моделей і методів підвищення гнучкості розкладу ІТ-проекту дистанційного навчання

1.4 Обґрунтування мети створення комбінованого методу та моделі підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання, формування вимог до них

Результати аналізу сучасного стану проблеми підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання дозволяють зробити такі висновки.

По-перше, впровадження ГРЗ підвищить прибуток та збільшить аудиторію студентів за рахунок того, що більша кількість людей зможе відвідувати курси. Крім того, підвищиться рівень конкурентоспроможності, студенти будуть в першу чергу дивитися на курси з гнучким розкладом. Також за рахунок кращого розподілення занять, фірма зможе вести більше курсів одночасно. Це, в свою чергу, також підвищить прибуток. ГРЗ доповнення до існуючих систем, тобто впровадження його не вимагає призупинення стадії ЖЦ експлуатації існуючого ІТ-проєкту.

По-друге, комбінований метод та модель підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання повинні зберегти плюси своїх аналогів таких як GreenForest, Gmail, Outlook та ascTimeTables, а також виправити їх недоліки, щоб досягти мети – доцільно розподіляти заняття курсу враховуючи побажання усіх їх учасників та наявні обмеження в автоматичному режимі [4, 5, 6, 7]. А саме:

- враховувати задані обмеження розкладу;
- враховувати побажання студентів;
- враховувати побажання викладачів;
- розміщувати заняття оптимально та в ті проміжки часу, що відповідають найкомпроміснішому з можливих.

По-третє, для створення гнучкого розкладу не має єдиного алгоритму, що можна було б розширювати додаючи обмеження під свої особливості

роботи. Також не має можливості додавати обмеження часу студентів, та розумно розподіляти заняття базуючись на них. В той же час більшість студентів дистанційних курсів потребує саме гнучкого розкладу як і викладачі.

Отже, комбінований метод та модель повинні акумулювати в собі сильні сторони існуючих методів шляхом використання їх на різних кроках формування розкладу. Тобто в основі ляже жадібний алгоритм, дані і умови для якого будуть отримані за допомогою алгоритму на основі обмежень та генетичного алгоритму. Проблему конфліктних та недостатніх даних вирішить ШНМ.

При цьому об'єднання розглянутих існуючих моделей та методів підвищення гнучкості розкладу ІТ-проекту дистанційного навчання компенсує їхні слабкі сторони й робить метод більш універсальним, тобто здатним вирішувати різноманітні задачі складання ГРЗ.

Опис проблем, які має вирішувати комбінований метод та модель підвищення гнучкості розкладу ІТ-проекту дистанційного навчання представлений нижче.

Першу проблему «Розстановка пріоритетів при створенні ГРЗ», яку повинен вирішувати комбінований метод, можна сформулювати так: проблема розстановки пріоритетів при створенні розкладу. Коріння цієї проблеми це пошук компромісного рішення для групи людей, що мають різні проміжки вільного часу. Задача методу обрати найбільший перетин цих проміжків, з урахуванням існуючих обмежень та вільного часу викладача. При цьому варто розподілити обмеження за пріоритетами. Наприклад, без викладача неможливо провести заняття, тобто можна вважати його доступність вищим пріоритетом, потім можуть йти вказані проміжки студентів, ще нижче налаштування тих хто не вказав час, й ще нижче ті в кого не має ні першого ні другого. Вирішення цієї проблеми за допомогою вказання чіткої ієрархії допоможе досягти найефективнішого перетину розподілу часу та задоволеності студентів і викладачів розкладом.

Створення гнучкого графіка для курсу дистанційного навчання на основі наявності часу у студентів вимагає ретельного планування та врахування різноманітних факторів. Планування включає в себе кілька кроків, що йдуть один за одним.

Опитування студентів, що проводиться для того щоб дізнатися, коли вони мають вільний час для занять. Надана інформація дозволить підлаштуватися під їх активності: робочий графік, сімейні та інші зобов'язання, які можуть вплинути на їхню здатність відвідувати заняття.

Визначення часових рамок на основі зібраних даних від студентів, треба визначити основні часові рамки, які підходять для більшості з них. Це може включати вечори, вихідні або інші блоки часу, коли студенти зазвичай доступні. Цей крок допоможе уникнути браку даних, а отримані проміжки допоможуть встановити налаштування для студента.

Друга проблема «Формулювання та корекція обмежень» до вирішення комбінованим методом: проблема необхідності визначення всіх обмежень та корекція їх у разі потреби. Найскладніший пункт у вирішенні цієї проблеми це аналіз існуючих процесів на фірмі, особливості проведення саме цього курсу, характер співпраці з викладачами. Також до цього можна віднести, визначення кількості занять протягом робочої неділі чи тижня, строків початку та кінця курсу. На основі цього можна буде підрахувати скільки занять вже проведено й скільки залишилося.

Третьою є проблема «Людська похибка», тобто проблема неповноти та неправильності інформації від студентів та викладачів. Необхідно розробити систему аналізу та підказок для вчителя і студентів при внесенні проміжків доступності. Наприклад, для викладача доцільно знати скільки уроків він може провести при надані даних про доступність та скільки потрібно враховуючи план його курсів.

Четверта проблема, а саме «Розклад повинен створюватися заздалегідь» полягає у тому, що створення розкладу повинно відбуватися

заздалегідь, це дасть час студентам скоригувати свій графік і гарантувати, що вони все ще зможуть брати участь у курсі.

Мета та, формально визначена, вимога методу створити гнучкий графік, який враховує вільний час студентів, водночас забезпечуючи їм комплексний та захоплюючий досвід навчання.

Критеріїв ефективності комбінованого методу підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання, нерозривно пов'язані з проблемами та вимогами до методу

Щоб протестувати комбінований метод підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання треба встановити яким критерієм ефективності він має задовольняти:

— оптимальність – метод повинен генерувати графіки, які є оптимальними або близькими до оптимальних з точки зору досягнення цілей планування;

— швидкість – метод має бути ефективним і здатним своєчасно генерувати розклади, особливо коли ви маєте справу з великими проблемами планування;

— масштабованість – метод повинен бути здатний вирішувати проблеми планування різного розміру та складності, а також бути здатним адаптуватися до мінливих вимог;

— адаптованість – метод повинен бути здатний враховувати зміни у вимогах і обмеженнях планування, а також нові завдання або ресурси, які можуть бути додані до проблеми планування;

— надійність – метод повинен бути здатний обробляти несподівані події або збої, такі як недоступність ресурсів або зміни в пріоритетах завдань;

— прозорість – метод має бути прозорим щодо того, як він генерує розклади, щоб користувачі могли зрозуміти, як працює метод і які фактори враховуються в процесі планування;

- зручність використання – метод має бути зручним і простим у використанні навіть для користувачів, які не знайомі з методами планування чи комп'ютерним програмуванням;
- точність – метод повинен створювати точні та надійні розклади, які не створюють конфліктів або помилок;
- узгодженість – метод має генерувати узгоджені розклади для різних прогонів і не повинен створювати різні розклади для тих самих вхідних даних;
- адаптивність – метод має бути адаптованим до різних середовищ і вимог планування, а також мати можливість вчитися на попередньому досвіді планування для покращення його продуктивності.

Розрахунок задоволеності графіком із найбільшим компромісом між вибраними сегментами можна підійти за допомогою багатокритеріальних методів прийняття рішень. Ці методи дозволяють розглядати кілька критеріїв або цілей при оцінці якості або задоволеності даного розкладу. Одним із популярних методів для цієї мети є аналітичний ієрархічний процес (АІП).

У АІП проблема прийняття рішення розбивається на ієрархію критеріїв, під-критеріїв і альтернатив. Особа, яка приймає рішення, призначає ваги кожному критерію на основі його відносної важливості та порівнює кожну альтернативу з кожним критерієм, щоб створити матрицю попарних порівнянь. Потім ці матриці попарного порівняння об'єднуються за допомогою математичного процесу, який називається розкладом на власний вектор, щоб створити загальний рейтинг або оцінку для кожної альтернативи. Матриці попарного порівняння комбінуються за допомогою розкладу на власний вектор для генерації загального рейтингу або оцінки для кожної альтернативи. Цей рейтинг можна використовувати для оцінювання графіка, чи пропонує він найбільший компроміс між вибраними сегментами, беручи до уваги відносну важливість кожного критерію та переваги особи, яка приймає рішення [12].

Оптимальність розкладу підраховується як відношення кількості занять, що потрібно розподілити, до кількості фактично розподілених занять за допомогою методу створення ГРЗ формула 1.1:

$$\text{Optimality} = \frac{Q_C}{Q_{CF}}, \quad (1.1)$$

де  $Q_C$  — загальна кількість занять;

$Q_{CF}$  — загальна кількість фактично розподілених занять.

Адаптивність методу до змін вхідних параметрів можна оцінити шляхом розрахунку чутливості або реагування методу на зміни цих параметрів.

Одним із способів обчислення показника адаптивності методу є використання формули 1.2, ця формула обчислює відношення діапазону можливих виходів до діапазону можливих входів, забезпечуючи міру того, наскільки метод реагує на зміни вхідних параметрів:

$$\text{Adaptability} = \frac{R_{output}}{R_{input}}, \quad (1.2)$$

де  $R_{output}$  — діапазон можливих виходів, які може створити метод, коли вхідні параметри змінюються;

$R_{input}$  — діапазон можливих вхідних параметрів, які метод може прийняти та обробити.

Стійкість методу до змін вхідних параметрів можна оцінити шляхом розрахунку стійкості методу до цих змін. Наприклад, якщо алгоритм планування має високу чутливість до змін вхідних даних, тобто невеликі варіації вхідних параметрів призводять до значних змін у вихідних даних, тоді алгоритм може бути не дуже надійним. Однак, якщо алгоритм має низьку чутливість до змін вхідних даних, що означає, що він виробляє

узгоджені результати навіть з варіаціями вхідних параметрів, тоді алгоритм є більш надійним формула 1.3:

$$Robustness = 1 - Adaptability \quad (1.3)$$

Одним із загальних показників повноти розподілу є коефіцієнт використання, який вимірює частку доступних ресурсів, які використовуються запланованими діями формула 1.4:

$$C_{use} = \frac{q_{slots}}{q_{sf}}, \quad (1.4)$$

де  $q_{sf}$  – кількість використаних слотів;

$q_{slots}$  – кількість заданих слотів.

Показник  $q_{slots}$  розраховується на основі вказаних проміжків доступності формула 1.5:

$$q_{slots} = \frac{((request_{end} - request_{start}) + B)}{D + B}, \quad (1.5)$$

де  $q_{slots}$  – кількість вільних слотів;

request – заявка на розклад;

$request_{end} - request_{start}$  – величина проміжку вказаного в заявці в хвиликах;

B – константна величина перерви в хвиликах;

D – константна величина уроку в хвиликах.

Співвідношення проблем та критеріїв ефективності наведено у таблиці 1.3.

Таблиця 1.3 – Співвідношення «Проблема» та «Критерій ефективності»

Проблема	Критерій ефективності	Спосіб обчислення
розстановка пріоритетів при створенні ГРЗ	оптимальність, точність, узгодженість	АП оптимальність розподілу занять формула 1.1
формулювання та корекція обмежень	прозорість, адаптованість	адаптованість методу формула 1.2 стійкість методу формула 1.3
людська похибка	надійність, зручність використання, адаптивність	показник повноти розподілу формула 1.4 та 1.5
розклад повинен створюватися заздалегідь	швидкість, масштабованість	підрахунок складності кожного з кроків алгоритму O, аналіз доцільності використання в різних масштабах

Отже, комбінований метод має вирішувати наведені проблеми «Розстановка пріоритетів при створенні ГРЗ», «Формулювання та корекція обмежень», «Людська похибка», «Розклад повинен створюватися заздалегідь», а також оцінювати свою роботу за допомогою критеріїв ефективності і їх числових показників. Ці критерії потрібні щоб переконатися, що метод гнучкого планування є ефективним і відповідає потребам користувачів.

## 1.5 Постановка задачі дослідження моделей і методів підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання

Об'єктом дослідження в рамках магістерської кваліфікаційної роботи є процес формування розкладу (6.5 Develop Schedule) ІТ-проєкту дистанційного навчання, згідно з Project Management Body of Knowledge's (PMBOK) [13].

Предметом дослідження являються моделі та методи аналізу вирішення задачі розкладу.

Метою даної роботи є дослідження моделей і методів підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання та створення комбінованого методу і моделі, які мають використовувати сильні сторони існуючих, мінімізуючи при цьому їхні недоліки.

Для досягнення мети, необхідно досліджувати наступні питання:

- проаналізувати існуючі методи розподілення робіт;
- обґрунтувати мету розробки комбінованого методу та сформулювати вимоги до комбінованого методу розподілення робіт;
- описати проблеми, які має вирішувати комбінований метод підвищення гнучкості розкладу;
- сформулювати вимоги до комбінованого методу підвищення гнучкості розкладу;
- сформулювати критерії ефективності до комбінованого методу підвищення гнучкості розкладу;
- розробити комбінований метод підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання;
- описати методіку застосування методу;
- застосувати комбінований метод на практиці задля відображення його переваг над іншими методами.

## **2 РОЗРОБКА КОМБІНОВАНОГО МЕТОДУ ТА МОДЕЛІ ПІДВИЩЕННЯ ГНУЧКОСТІ РОЗКЛАДУ ІТ-ПРОЄКТУ ДИСТАНЦІЙНОГО НАВЧАННЯ**

2.1 Опис комбінованого методу підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання, етапи створення розкладу

Ґрунтуючись на аналізі актуальності дослідження, аналізі існуючих рішень, аналізі існуючих моделей та методів, та обґрунтуванні мети створення комбінованого методу, був розроблений комбінований метод підвищення розкладу ІТ-проєкту дистанційного навчання. Створення розкладу комбінованим методом буде проходити в декілька етапів, кожен з яких буде мати під собою адаптований існуючий алгоритм чи їх поєднання: жадібний алгоритм, алгоритм на основі обмежень, генетичний алгоритм. Для корекції та доповнення даних буде використана ШНМ.

Комбінований метод підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання складається із трьох основних етапів, які являються взаємопов'язаними – вхідними даними наступного етапу є вихідні дані попереднього. Кожен етап має на меті певну ціль: підготовку даних, створення обмежень, перебір варіантів, тощо. Отже, комбінований метод підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання має наступні етапи: формування обмежень, підготовка вхідних даних, формування розкладу: пріоритизація завдань, формування розкладу: обробка конфліктів.

Кожен з етапів має свої вхідні дані, при цьому кожен наступний етап використовує результати попереднього, в реалізації етапів задіяні: ШНМ прогнозування проміжків часу студентів та викладачів, алгоритм на основі обмежень, генетичний алгоритм та жадібний алгоритм. Алгоритми використовуються в поєднанні та адаптуються під комбінований метод підвищення гнучкості розкладу.

Схематичне зображення етапів створення розкладу за комбінованим методом подано на рис. 2.1.



Рисунок 2.1 – Схематичне зображення етапів створення розкладу за комбінованим методом

Етап «1. Формування обмежень» відповідає за створення списку обмежень системи за її налаштуваннями. На основі списку обмежень створюється список критеріїв значущості і їх ваг. З урахуванням цих списків відбувається створення ієрархічної структури згідно з пріоритетами, окрім її останнього рівня. Крім того, на основі обмежень, формуються умови завершення ітерацій жадібного вибору та умова оцінювання успішного створення розкладу. Вихідні дані: ієрархічна структура та сформовані умови, будуть використовуватися наступними етапами.

Етап «2. Підготовка вхідних даних на основі вводу від користувачів»

відповідає за формування даних, що є базовими для створення розкладу і представляють собою останній рівень ієрархічної структури. Кінцева мета етапу створити згрупований однорідний список сутностей проміжків вільного часу, з введених даних від користувачів, тобто студентів та викладачів. За відсутності даних від користувачів, чи недостатньому ввході, використовується ШНМ для генерування даних наближених до реальних.

Етап «3. Формування розкладу», складається з двох під-етапів: «3.1. Пріоритизація завдань» та «3.2. Обробка конфліктів».

Під-етап «3.1. Пріоритизація завдань» керується результатами алгоритму на основі обмежень, а саме ієрархічною структурою даних, формує ітерації створення розкладу, шляхом рекурсивного проходу по кожному з рівнів від кореневого до останнього.

Під-етап «3.2. Обробка конфліктів» є кінцевою віхою формування розкладу, на цьому етапі йде підбиття підсумків та отримання результатів. Кожна з повних ітерацій «3.1. Пріоритизація завдань» має критерії виходу, а остання – крім того оцінює чи досягнута мета створення розкладу. У разі виникнення конфліктів застосовується доповнення проміжків, ітерація формування розкладу починається з початку.

Отже, комбінований метод на кожному етапі використовує сильні сторони алгоритмів, компенсуючи їх недоліки та слабкі сторони.

По-перше, етап «1. Формування обмежень» використовує сильні сторони алгоритму на основі обмежень, а саме, дає змогу жадібному алгоритму опрацьовувати велику кількість обмежень розкладу. При цьому формальне описання обмежень, зведення їх до критеріїв значущості і ваг робить алгоритм на основі обмежень простим в реалізації, нівелюючи його основну ваду.

По-друге, етап «2. Підготовка вхідних даних на основі вводу від користувачів» спирається на властивість генетичного алгоритму працювати з великою кількістю завдань. Так як, йдеться про дані отриманні від користувачів на певний період, покоління не є перевантаженими і

опрацювання навіть великої кількості критично не сповільнить роботу всього комбінованого методу в цілому. В свою чергу, сформовані дані нівелюють ваду жадібного алгоритму з обробкою тільки маленької кількості завдань. Усі розглянуті в роботі алгоритми не можуть працювати в умовах браку даних, тому, для доповнення даних наближеними до реальних, на цьому етапі використовується ШНМ.

По-третє, етап «3. Формування розкладу» та його під-етапи реалізують створення розкладу на основі жадібного алгоритму. Як відомо, цей алгоритм видає найбільш ефективні рішення на добре пріоритизованих списках. Ієрархічна структура, отримана з попередніх етапів, і дає змогу використати плюси алгоритму: простоту в реалізації та якнайшвидше отримання готового розкладу. Під-етап «3.2. Обробка конфліктів», в свою чергу, користується ШНМ для доповнення користувацьких даних.

## 2.2 Використання штучної нейронної мережі для розробки комбінованого методу підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання

Передбачити дату й час на основі існуючих алгоритмів машинного навчання можна за допомогою регресійних моделей. Регресійні моделі використовують історичні дані для прогнозування майбутніх результатів. У разі прогнозування дати й часу регресійні моделі можна використовувати для аналізу закономірностей у даних і прогнозування майбутніх дат і часу [14].

Щоб створити регресійну модель, історичні дані спочатку збираються, а потім розбиваються на набори для навчання та тестування. Навчальний набір використовується для навчання моделі, а тестовий набір використовується для оцінки точності моделі. Коли модель навчена, її можна використовувати для прогнозування майбутніх дат і часу [14].

Регресійні моделі також можуть бути реалізовані в контексті штучних нейронних мереж. Нейронні мережі – це тип алгоритму машинного навчання, який, зокрема, можна використовувати для завдань регресії.

У регресійній моделі нейронної мережі вхідні дані спочатку обробляються одним або кількома шарами нейронів, кожен з яких застосовує до вхідних даних математичну функцію. Вихідні дані кожного шару потім передаються на наступний рівень, доки остаточний вихідний рівень не створить прогнозовані значення.

Процес навчання регресійної моделі нейронної мережі передбачає коригування вагових коефіцієнтів зв'язків між нейронами, щоб мінімізувати різницю між прогнозованими значеннями та фактичними значеннями в навчальних даних. Зазвичай це робиться за допомогою алгоритму оптимізації.

ШНМ для комбінованого методу підвищення розкладу буде мати два прихованих шари, кількість вхідних сигналів-параметрів дорівнюватиме кількості атрибутів сутності «Побажання щодо створення розкладу». Тобто за правилом – кількість нейронів у вхідному шарі дорівнює кількості ознак у даних.

Кількість прихованих нейронів має становити  $2/3$  розміру вхідного шару плюс розмір вихідного шару, тобто нерівність, кількість прихованих нейронів має бути менше ніж подвійний розмір вхідного шару, повинна виконуватись формула 2.1 та формула 2.2:

$$p = \frac{2 * n}{3} + 1, \quad (2.1)$$

$$p < 2n, \quad (2.2)$$

де  $p$  – кількість прихованих нейронів;

$n$  – кількість вхідних сигналів-параметрів.

Кількість нейронів у вихідних даних залежить від того, чи використовується модель як регресор чи класифікатор. В нашому випадку модель вирішує задачу регресії і має відповідно один вихід у вигляді часового проміжку формула 2.3:

$$[[start_i, end_i], \dots [start_n, end_n]], \quad (2.3)$$

де  $start$  – початок часового проміжку у вигляді десяткового дробу;

$end_i$  – кінець часового проміжку.

Мережа (network) складається зі з'єднань (connection), кожне з яких передає вихід нейрону  $i$  до входу нейрону  $j$ . В цьому сенсі  $i$  є попередником (predecessor)  $j$ , а  $j$  є наступником (successor)  $i$ . Кожному з'єднанню призначено вагу (weight)  $w_{ij}$  [15].

Вагові коефіцієнти нейронної мережі вивчаються в процесі навчання. Мережа починається з випадково ініціалізованих ваг, а потім ваги коригуються ітеративно, щоб мінімізувати функцію втрат. Функція втрат є мірою того, наскільки добре мережа працює з навчальними даними, і метою є її мінімізація [15].

Під час навчання мережа робить прогнози на основі навчальних даних, а функція втрат обчислюється на основі різниці між прогнозованим виходом і фактичним виходом. Потім ваги коригуються за допомогою алгоритму оптимізації, такого як градієнтний спуск, щоб мінімізувати втрати.

У міру коригування ваги мережа вчиться робити кращі прогнози на основі навчальних даних. Після завершення навчання мережу можна використовувати для прогнозування нових, невідомих даних.

В якості функції активації обрано ReLU (Rectified Linear Unit), вона зазвичай використовується в нейронних мережах, у тому числі для завдань регресії, оскільки вона ефективна з точки зору обчислень і може допомогти запобігти проблемі зникнення градієнта [16].

Проблема зникнення градієнта виникає, коли градієнт (нахил функції втрат відносно ваг) стає дуже малим під час навчання, що може ускладнити навчання нейронної мережі та зближення до оптимального рішення. Функція ReLU допомагає вирішити цю проблему, не даючи градієнту стати занадто малим у додатній області функції, дозволяючи мережі навчатися ефективніше [16].

Крім того, функція ReLU добре підходить для завдань регресії, оскільки вона створює вихідні значення в діапазоні  $[0, \infty)$ , який є звичайним діапазоном вихідних даних для задач регресії. Це означає, що функція ReLU може забезпечити хорошу підгонку даних у багатьох випадках, що робить її популярним вибором для регресійних нейронних мереж.

У якості функції втрат (loss function), використано Adam – це популярний алгоритм оптимізації, який використовується для навчання нейронних мереж, особливо для глибокого навчання. Він поєднує в собі переваги двох інших алгоритмів оптимізації: AdaGrad (Adaptive Gradient Algorithm) і RMSProp (Root Mean Squared Propagation) [17].

У контексті регресійних нейронних мереж оптимізатор Adam є хорошим вибором з кількох причин [17]:

- адаптивна швидкість навчання: швидкість навчання автоматично регулюється для кожної ваги в нейронній мережі на основі попередніх градієнтів. Це допомагає моделі швидше сходитися та уникнути застрягання в локальних мінімумах;

- імпульс: оптимізатор використовує імпульс для прискорення конвергенції (процес зближення, сходження компромісів), що допомагає згладити шум у градієнтах і забезпечити більш стабільні оновлення;

- регуляризація: оптимізатор Adam включає термін регуляризації, який допомагає запобігти переобладнанню, це особливо важливо в задачах регресії, де модель може легко перепідлаштуватися під навчальні дані.

— ефективно використання пам'яті: потребує менше пам'яті порівняно з іншими алгоритмами оптимізації, такими як AdaGrad, що робить його придатним для великих наборів даних.

Таким чином, оптимізатор Adam є хорошим вибором для регресійних нейронних мереж, оскільки він забезпечує швидку конвергенцію, краще узагальнення та ефективно використання пам'яті.

Архітектура ШНМ для етапів «2. Підготовка вхідних даних на основі вводу від користувачів» та «3.2. Обробка конфліктів» складання розкладу за комбінованим методом підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання наведена на рис. 2.2.

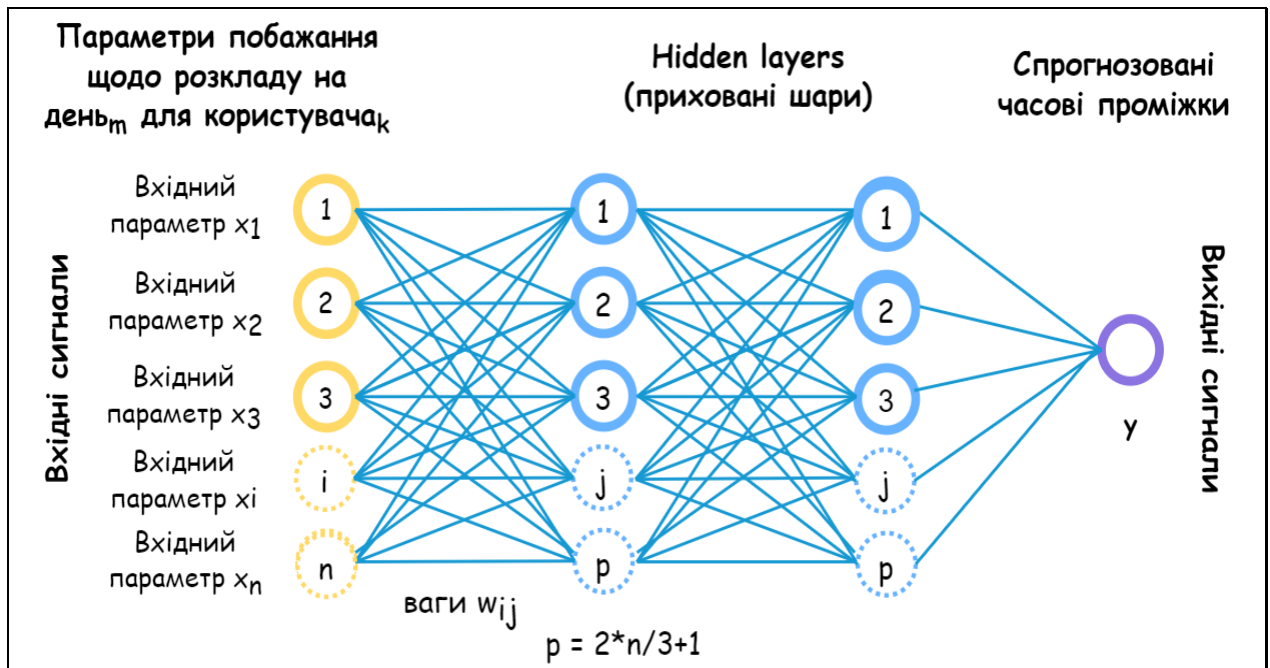


Рисунок 2.2 – Архітектура штучної нейронної мережі для прогнозування часу занять

Отже, розроблена ШНМ використовується на двох етапах «2. Підготовка вхідних даних на основі вводу від користувачів» та «3.2. Обробка конфліктів» створення розкладу і, за допомогою доповнення вхідних даних від користувачів, даними наближеними до реальних, робить можливим створення оптимального компромісного розкладу занять з

урахуванням побажань як від студентів, так і від викладачів курсів згідно з рис. 2.3.



Рисунок 2.3 – Схематичне зображення використання ШНМ етапами комбінованого методу підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання

Розроблена ШНМ комбінованого методу підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання має наступні характеристики:

- два прихованих шари;
- кількість вхідних сигналів дорівнює  $n$ , тобто кількості ознак у вхідних даних;

- кількість прихованих нейронів знаходиться за формулою 2.1 та формулою 2.2;
- кількість нейронів у вихідних даних дорівнює одному у вигляді часового проміжку формула 2.3;
- функція активації ReLU;
- функція втрат Adam.

2.3 Використання алгоритму на основі обмежень для розробки комбінованого методу підвищення гнучкості розкладу IT-проєкту дистанційного навчання

Алгоритм на основі обмежень використовується у комбінованому методі підвищення гнучкості розкладу IT-проєкту дистанційного навчання задля вирішення проблеми визначення всіх обмежень. Ґрунтуючись на результатах визначення обмежень, відбувається виведення критеріїв значущості та загальних умов вибору та пріоритизації.

Перший етап формування розкладу це «1. Формування обмежень». Завдання виконання обмежень (Constraint satisfaction problem) (ЗВО) – це математичні проблеми, визначені як сукупність об'єктів, стан яких має задовольняти ряду обмежень. ЗВО представляє сутності проблеми як однорідний набір обмежень, що накладаються на змінні, які розв'язуються методами виконання обмежень [18].

ЗВО є предметом інтенсивних досліджень і в галузі штучного інтелекту, і дослідження операцій, оскільки закономірності у формулюванні цих завдань складають загальну основу для аналізу та вирішення проблем у багатьох неспоріднених областях. ЗВО часто мають високу складність, що

вимагає поєднання евристичних та комбінаторних методів пошуку для швидкого вирішення [18].

Згідно з ЗВО, формально сутність проблеми визначається трійкою  $(X, D, C)$ ,  $X$  – множина змінних,  $D$  – область значень, а  $C$  – множина обмежень [19].

Множина  $C$  представляє собою набір обмежень  $C_1(S_1)..C_n(S_n)$ , де  $S_i$  – набір значень, а обмеження  $C_i$  – це комбінація допустимих значень для змінних  $S$  [19].

Обмеження розкладу для комбінованого методу поділяються на обов'язкові – за замовчуванням, та необов'язкові – тобто такі, що залежать від особливостей роботи кожної конкретної фірми і можуть адаптуватися під її нужди.

Визначення обов'язкових обмежень наведено у вигляді таблиці 2.1.

Таблиця 2.1 – Формальне визначення завдання виконання обмежень для комбінованого методу підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання

$X$ – множина змінних	$D$ – область значень	$C$ – множина обмежень
навантаження викладача	$0 \leq$ кількість хвилин на тиждень $\leq$ максимальна кількість годин на тиждень	{0, 1 ...максимальна кількість годин на тиждень}
тривалість уроку	$0 <$ тривалість уроку $\leq 24 * 60$	{тривалість уроку}
тривалість перерви викладача	$> 0$ , в хвиликах, годинах, днях і т.д.	{тривалість перерви}
період часу, на який створюється розклад	період часу, що належить відрізьку, де початок – найбільш рання дата початку з курсів для яких складається розклад, кінець – найпізніша дата закінчення	тиждень, місяць і т.д.

Кінець таблиці 2.1

X – множина змінних	D – область значень	C – множина обмежень
кількість занять курсу на період	$0 \leq$ кількість занять курсу на період $\leq$ загальна кількість занять	{кількість занять курсу на період}
кількість проведених занять курсу на період	$0 \leq$ кількість проведених занять курсу на період $\leq$ загальна кількість занять курсу на період	{кількість проведених занять курсу на період}
кількість вільних слотів користувача на період по днях	для днів: сумарна кількість днів для курсів на період $\leq$ кількість заданих днів $\leq$ максимальна кількість днів в періоді	кількість вільних слотів користувача на період по днях

Другим етапом складання розкладу є «2. Підготовка вхідних даних на основі вводу від користувачів», до задачі формування даних також відноситься їх перевірка на валідність.

Перевірки на валідність включають себе наступні:

- перевірка кількості слотів на період;
- перевірка кількості днів на період в поданих побажаннях щодо розкладу.

Ці перевірки необхідні для контролю можливості створення розкладу з вхідних даних. Аналогічна перевірка виконується й для умови виходу з жадібного перебору.

Третій етап формування розкладу, а саме, його під-етап «3.1. Пріоритизація завдань» також використовує обмеження для обчислення ваги кожного курсу та проміжку за стандартним сценарієм, виходячи з яких формується умова, що виконується при порівнянні на основі їх параметрів

зادля проведення сортування та створення упорядкованих списків  
рисунок 2.4.

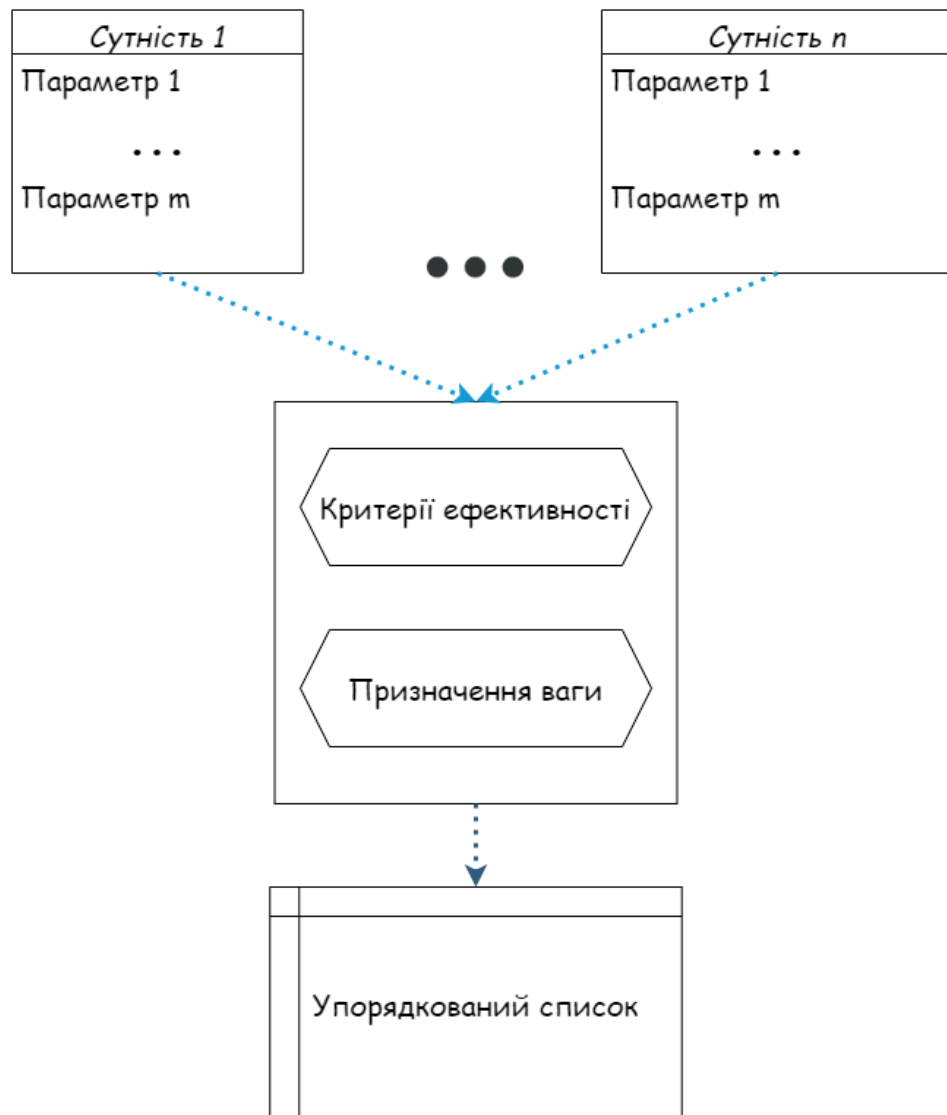


Рисунок 2.4 – Схематичне зображення процесу створення упорядкованого списку

Тобто, щоб провести сортування списків: з параметрів кожної сутності вибираються значущі (тобто такі які впливають на пріоритет), потім за допомогою критеріїв прораховуються ваги. Ці параметри можуть відрізнитися в залежності від конкретної реалізації методу.

Критерії значущості розраховуються на основі заданих обмежень та їх пріоритетів.

Критерії значущості, на основі обов'язкових обмежень та стандартного набору параметрів проміжку вільного часу, для комбінованого методу підвищення гнучкості розкладу ІТ-проекту дистанційного навчання представлено в таблиці 2.2.

Таблиця 2.2 – Приклад задання критеріїв значущості, їх ваг

Тип сутності	Назва параметру	Критерій значущості	Вага
курс	кількість занять на тиждень	кількість занять на тиждень → тиждень	2
проміжок вільного часу користувача (згрупований)	кількість співпадінь	кількість співпадінь вподобань → кількість студентів курсу + викладач	1

Тобто, виходячи з даних таблиці 2.2. – для стандартних обмежень розкладу, спочатку йде групування по курсах з значенням ваги 2, потім для кожного з курсів, наступним рівнем, буде йти групування за проміжками вільного часу користувачів з значенням ваги меншим 1.

Формування проміжків вільного часу користувача (згрупованих) відбувається на другому етапі створення розкладу «2. Підготовка вхідних даних на основі вводу від користувачів».

При цьому сортування буде йти згідно з критеріями значущості: на верх списку потрапляють курси з найвищим значенням кількості занять, а для списку проміжків – ті, що мають найбільшу кількість співпадінь вподобань.

Розширити реалізацію комбінованого метода можна за допомогою додавання обмежень і параметрів за таким же алгоритмом дій.

Кінцевим результатом роботи алгоритму на основі обмежень буде ієрархічна структура з сутностей, пов'язаних з розкладом, де кожен рівень представляє собою упорядкований список. При цьому коренева сутність буде

на першому рівні, а наступні будуть розташовані за порядком зменшення ваг (рисунок 2.5).

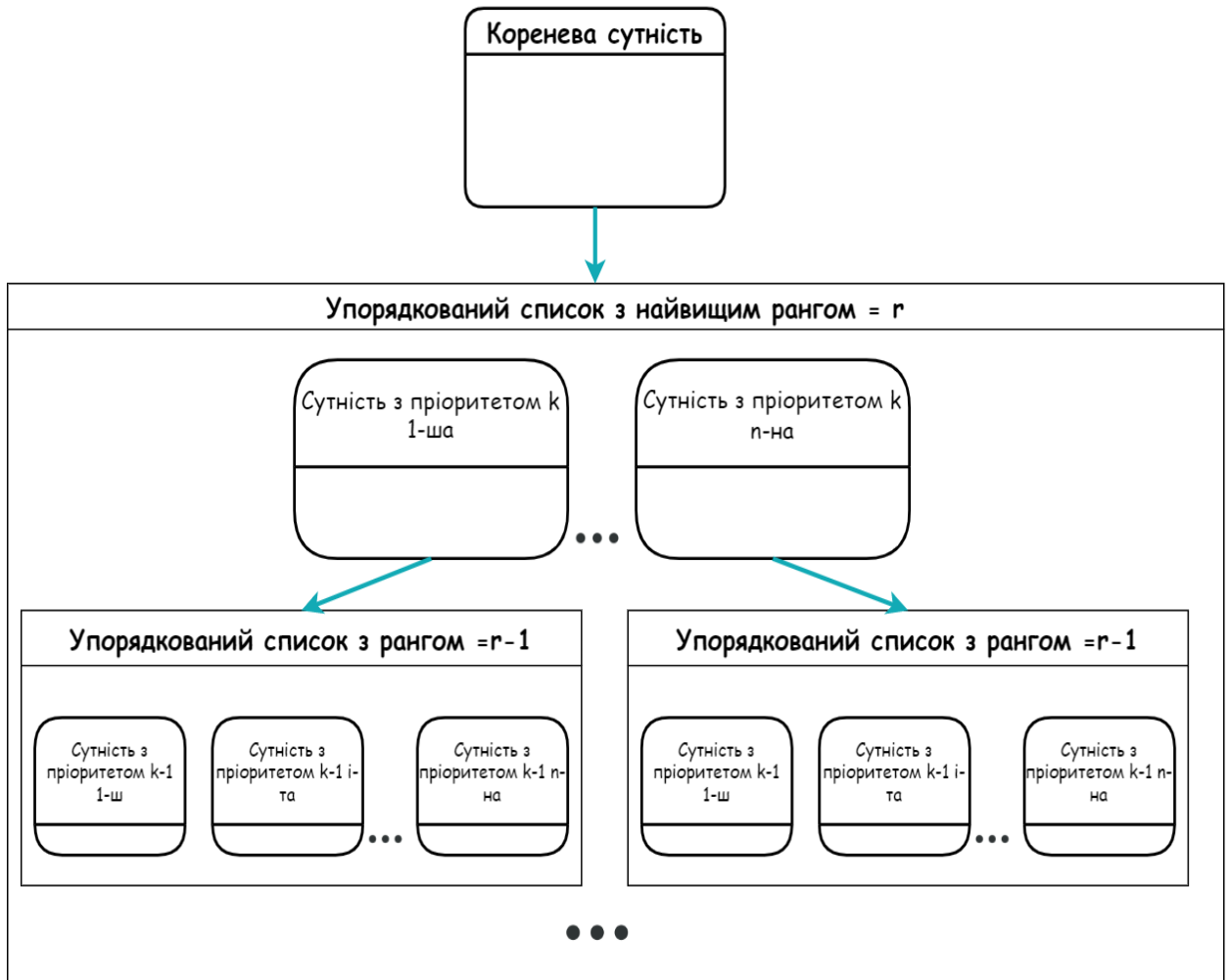


Рисунок 2.5 – Результат алгоритму на основі обмежень для етапу «Формування розкладу: пріоритизація завдань»

На під-етапі формування розкладу, «3.2. Обробка конфліктів», виконується перевірка розкладу: а саме, порівняння кількості запланованих занять методом та кількістю занять, що потрібно було запланувати. Коли ці два значення збігаються ітерація завершується, якщо ні – то, за стандартним сценарієм, йде аналіз, яких слотів не вистачило студентів чи викладача. Та згідно цього чи ШНМ додає слоти викладача, чи доповнення відбувається з незадіяного ще часу.

Отже, алгоритм на основі обмежень використовується на всіх етапах створення розкладу, вихідними даними цього алгоритму є:

- список обмежень;
- список критеріїв значущості та їхніх ваг;
- ієрархічна структура (крім останнього рівня);
- умови оцінювання валідності проміжків часу;
- умови оцінювання завершення ітерації;
- оцінювання успішного створення розкладу.

Приклад використання результатів алгоритму етапами комбінованого методу подано на рис 2.6.



Рисунок 2.6 – Використання результатів алгоритму на основі обмежень етапами комбінованого методу

## 2.4 Використання генетичного алгоритму для розробки комбінованого методу підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання

Генетичний алгоритм використовується у комбінованому методі на етапі «Формування розкладу: пріоритизація завдань».

Першим кроком на цьому етапі буде створення початкової популяції. Вид сутностей, з яких буде генеруватися розклад, повинен належати найнижчому рівню ієрархічної структури з алгоритму на основі обмежень. А саме формування популяції повинно передувати її створенню.

Найнижчий рівень сутностей – це згруповані й злиті проміжки вільного часу. Далі, розглядається послідовність дій на основі цього сценарію, при умові базового набору параметрів.

У генетичному алгоритмі початкова популяція відноситься до початкового набору потенційних рішень розв’язуваної проблеми, тобто, з точки зору комбінованого методу, створенню списку найнижчого рівня для ієрархічної структури.

Комбінований метод підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання створює початкову популяцію на основі експертної оцінки (така популяція, являється адаптованою до поточної проблеми), а саме на основі заданих студентами вільних проміжків часу.

У генетичному алгоритмі «хромосома» – набір, що визначає запропоновані параметри можливого рішення, а «ген» – це одна з «букв» строки «хромосоми», чи параметр, характеристика [20].

Кросовер (crossover) – це одна з ключових операцій у генетичному алгоритмі, що забезпечує змішування генетичної інформації між батьківськими особинами з метою створення нащадку. Він аналогічний біологічному кросоверу. Тобто, як і у біологічному кросоверу, процес кросовера відбувається наступним чином: як вхідні дані вибираються кілька

батьків, і один або кілька потомків створюються з використанням їх генетичного матеріалу [20].

Приклад хромосоми початкової популяції комбінованого методу підвищення гнучкості розкладу наведено на рисунку 2.7.

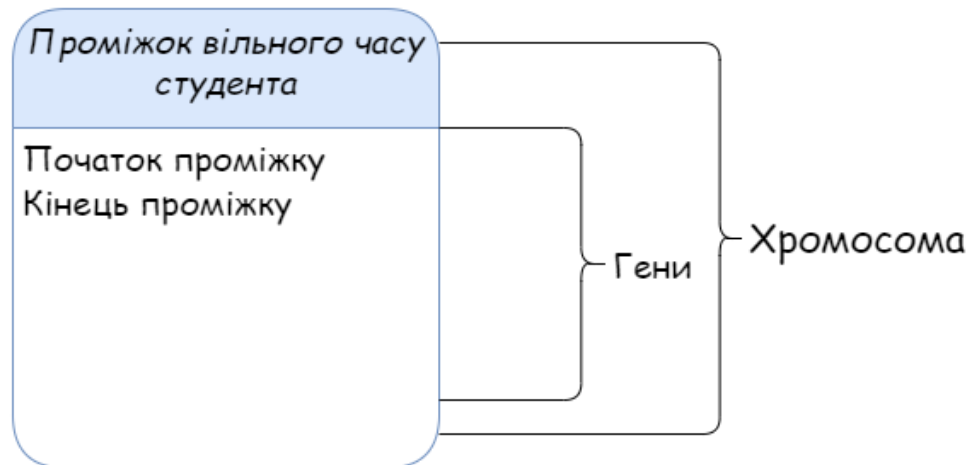


Рисунок 2.7 – Приклад хромосоми початкової популяції комбінованого методу

Друге покоління формується з початкового до якого за допомогою генетичного оператора мутації додаються проміжки для студентів, що не задали взагалі на цей період жодного проміжку. Створенням додаткових проміжків займається ШНМ.

Друге покоління фільтрується базуючись на проміжках викладача, всі проміжки викладача також додаються до розкладу, ті проміжки часу студентів, що не мають пересічень з викладацькими, прибираються. При цьому мається на увазі, що викладач задає проміжків більше ніж потрібно для розташування всіх його занять впритул.

Третє покоління формується за допомогою генетичного оператора кросовера. Хромосоми другого покоління зливаються за наступним принципом: кожний проміжок що належить іншому зливається за принципом найпізніше начало, найраніший кінець. На цьому ж рівні йде фільтрація: всі проміжки, що підпадають під вже створені заняття відбраковуються.

Алгоритм злиття проміжків комбінованого методу підвищення гнучкості розкладу ІТ-проекту дистанційного навчання наведено на рис. 2.8.

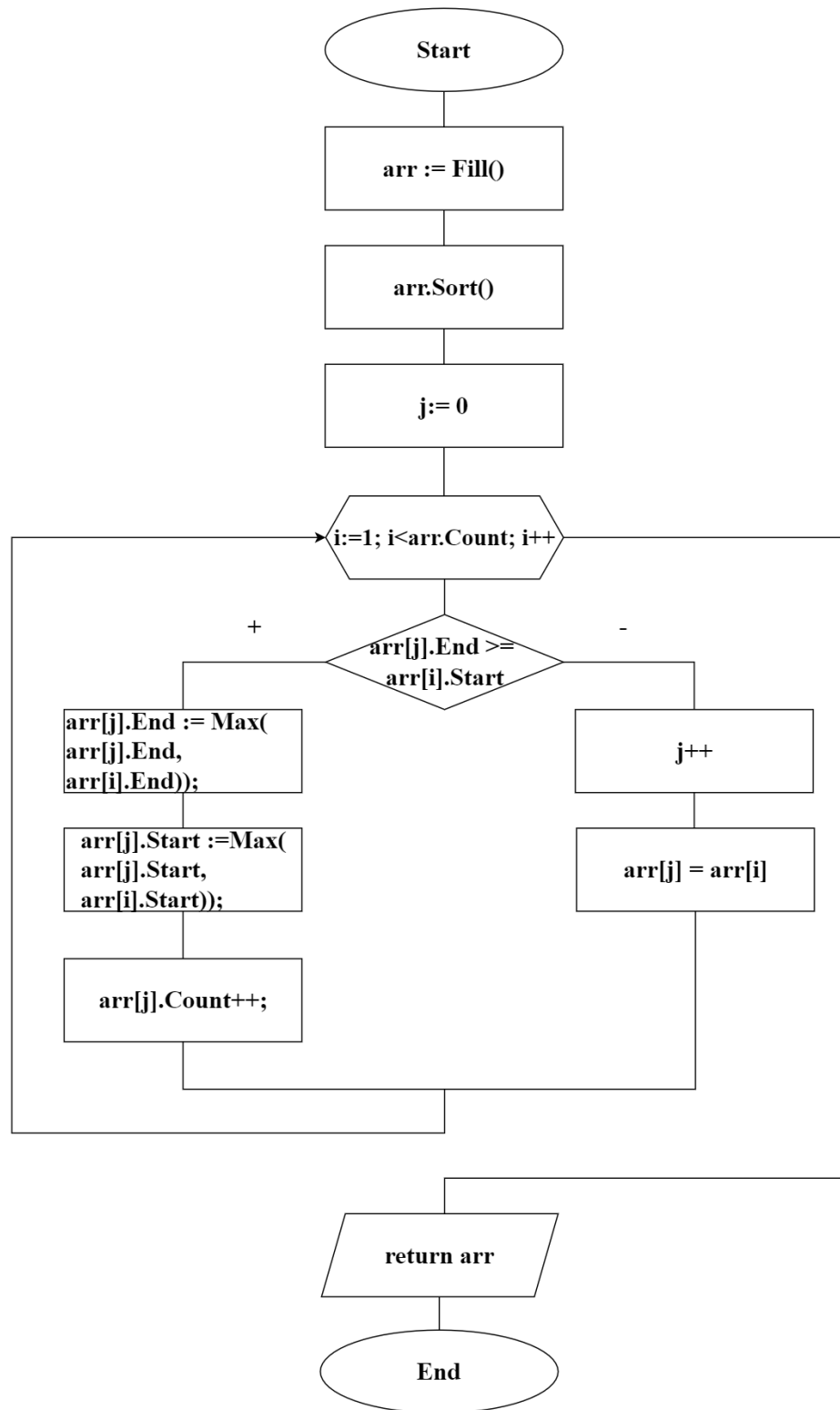


Рисунок 2.8 – Алгоритм злиття проміжків комбінованого методу

Фільтрація співпадінь з вже створеними заняттями має нюанси в роботі, а саме при видаленні співпадінь треба враховувати існуючі обмеження. Так в базовому варіанті це – тривалість уроку, тривалість перерви викладача.

Тобто, віднімаючи від проміжків частину спільну з існуючим заняттям, треба виходячи з того, що початок і кінець треба змістити на величину перерви. Відфільтровані проміжки повинні дорівнювати чи бути довшими за задану тривалість уроку (рис. 2.9).

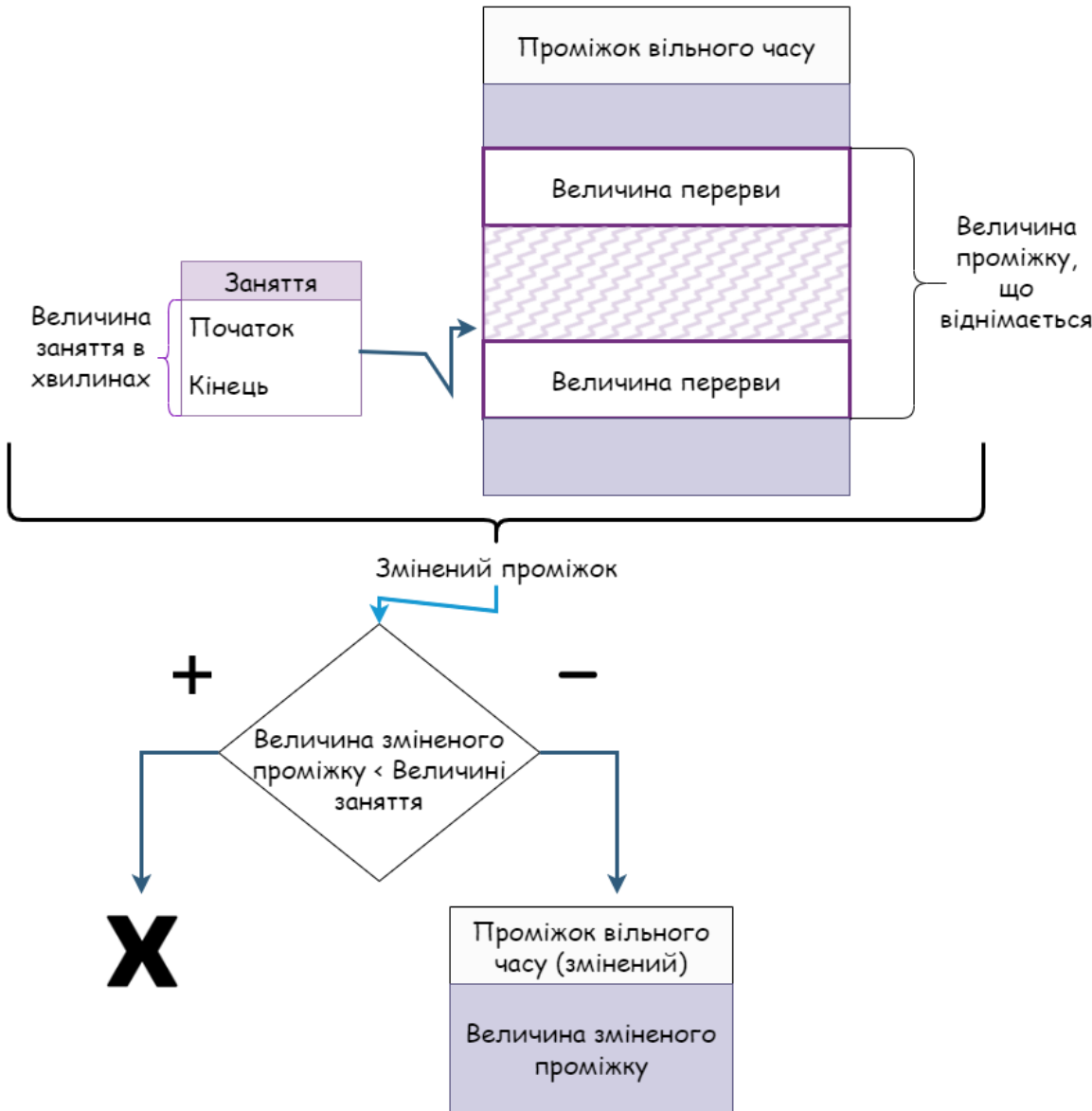


Рисунок 2.9 – Фільтрація співпадінь частин проміжків вільного часу з вже створеними заняттями

Підводячи підсумок, генетичний алгоритм у комбінованому методі використовується задля створення останнього рівня ієрархічної структури для алгоритму на основі обмежень (рис. 2.10).

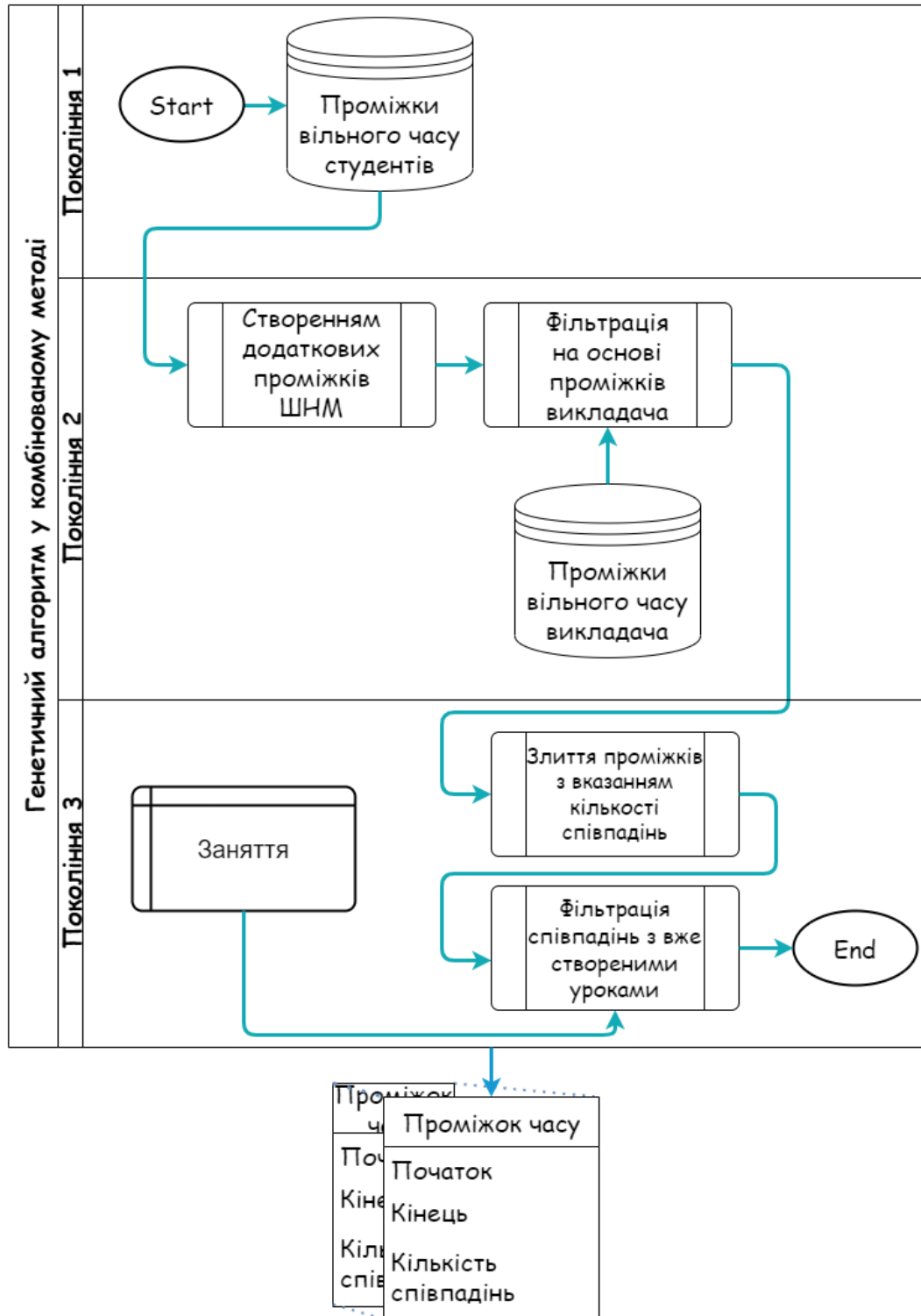


Рисунок 2.10 – Використання генетичного алгоритму у комбінованому методі підвищення гнучкості розкладу

Отже, результат генетичного алгоритму – це згруповані проміжки вільного часу користувачів, з вказанням кількості співпадінь вподобань. Список проміжків упорядковано за критерієм значущості «Кількість співпадінь» в низхідному порядку.

Цей список являється результатом другого етапу комбінованого методу, а саме етапу «2. Підготовка вхідних даних на основі вводу від користувачів».

Схематичне зображення використання другим етапом комбінованого розкладу результату генетичного алгоритму наведено на рис. 2.11.



Рисунок 2.11 – Схематичне зображення використання етапом комбінованого розкладу результату генетичного алгоритму

## 2.5 Використання жадібного алгоритму для розробки комбінованого методу підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання

Жадібний алгоритм використовується у комбінованому методі підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання задля створення розкладу, так званим, «жадібним вибором».

Жадібний вибір використовується в різних задачах оптимізації, включаючи планування, він добре працює для правильно пріоритизованих структур. У якості структури із правильним пріоритетом у комбінованому методі виступає ієрархічна структура. Під-етап формування розкладу комбінованого методу «Пріоритизація завдань» ґрунтується на ієрархічній структурі і представляє собою рекурсивний прохід по ній (рис. 2.12).

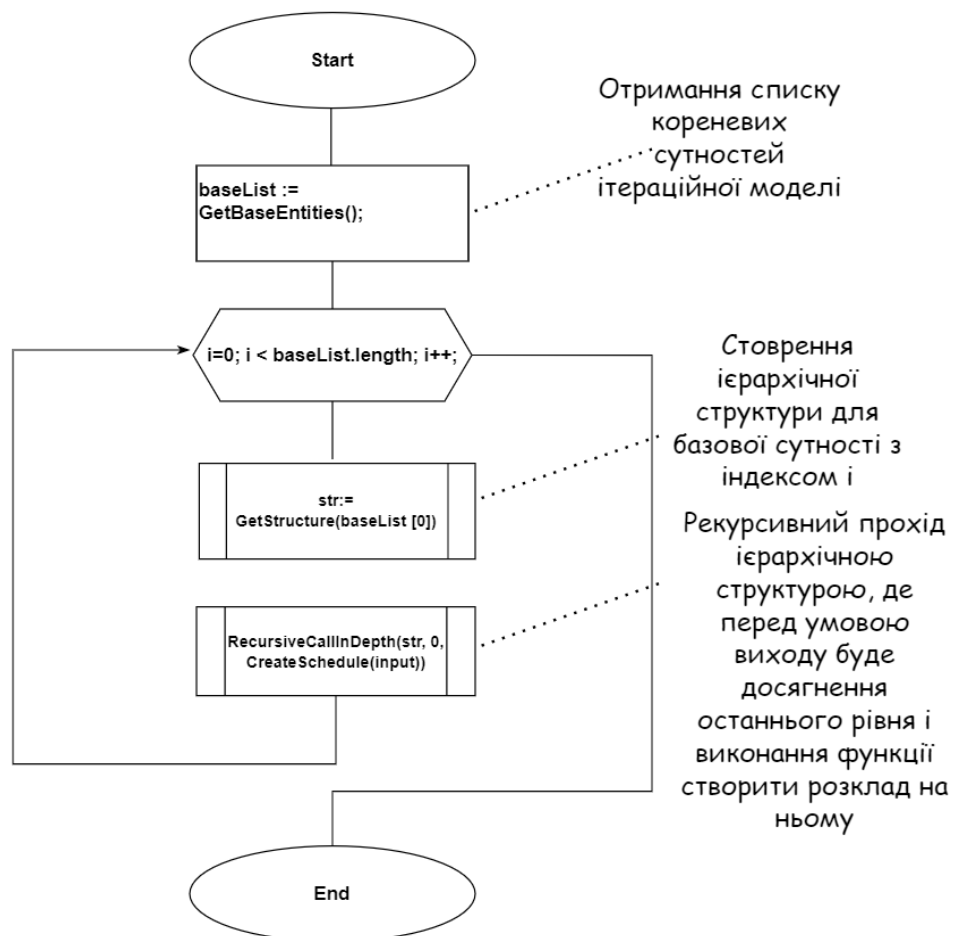


Рисунок 2.12 –Алгоритм проходу по ієрархії сутностей розкладу

Алгоритм створення розкладу починається з перевірки на відповідність запланованої кількості слотів до необхідної. У разі нестачі, визивається функція усунення конфліктів, що доповнює проміжки. В іншому випадку – циклічно створюється заняття поки їх кількість не зрівняється з заданою бажаною (рис. 2.13).

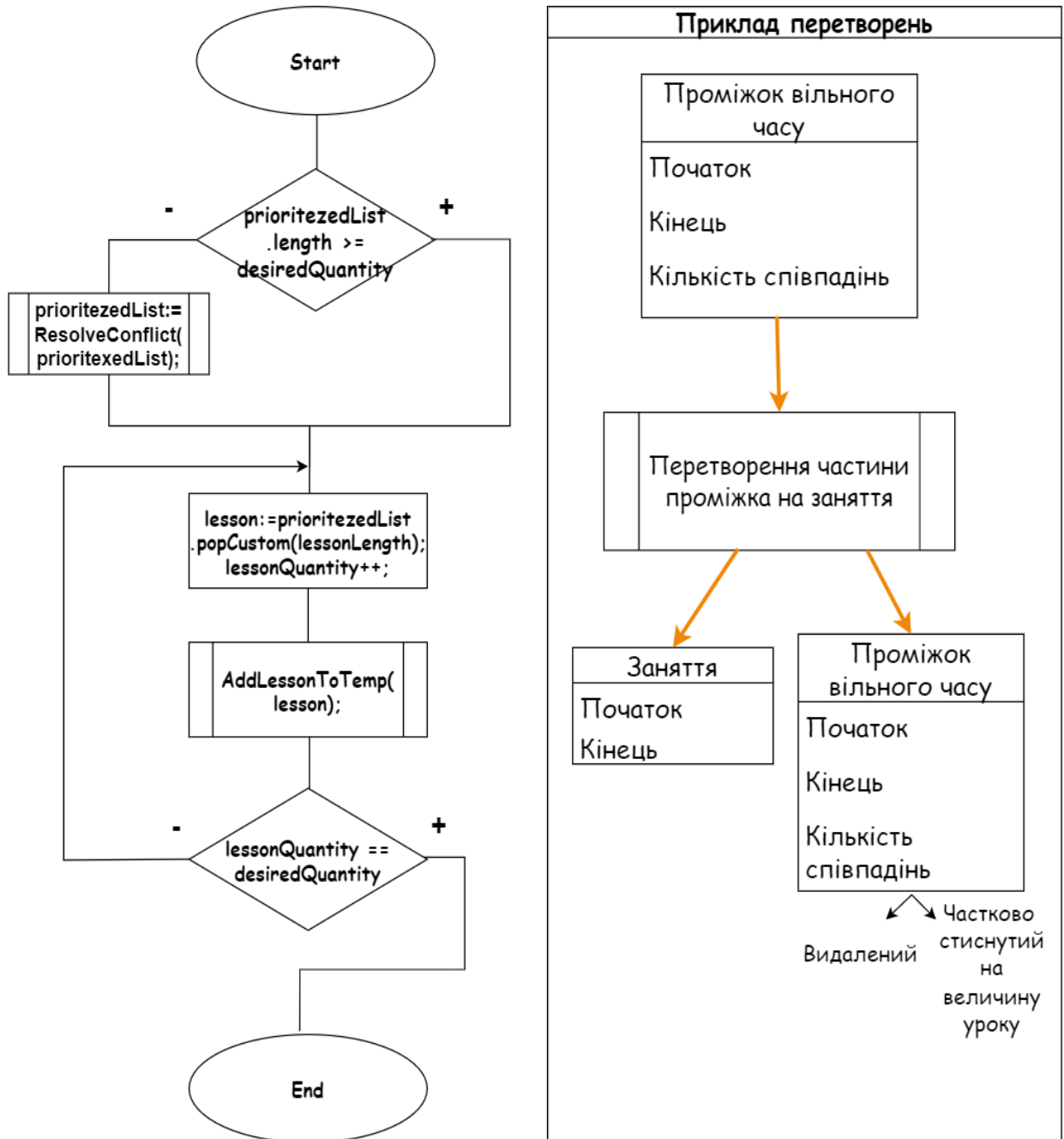


Рисунок 2.13 – Алгоритм функції створення розкладу з прикладом перетворень

Кінцевим результатом комбінованого методу є сутності «Заняття», з яких і складається розклад. Під-етап формування розкладу «3.2. Обробка конфліктів» відповідає за створення розкладу, аналіз успішності ітерації та усунення проблем.

Схематичне зображення використання результатів попередніх етапів жадібним алгоритмом та його взаємозв'язок з іншими алгоритмами показано на рис. 2.14.



Рисунок 2.14 – Схематичне зображення використання результатів попередніх етапів жадібним алгоритмом та його взаємозв'язок з іншими алгоритмами

Отже, жадібний вибір відбувається на основі результатів інших алгоритмів: алгоритму на основі обмежень, генетичному алгоритмі та моделі ШНМ. Жадібний алгоритм використовується третім етапом комбінованого методу «3. Формування розкладу» та його двома під-етапами «3.1. Пріоритизація занять», «3.2. Обробка конфліктів». Його результатом роботи є список занять, тобто розклад.

## 2.6 Висновки з розробки комбінованого методу підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання

Грунтуючись на аналізі актуальності дослідження, аналізі існуючих рішень, аналізі існуючих моделей та методів, та обґрунтуванні мети створення комбінованого методу – був розроблений комбінований метод підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання.

Комбінований метод має 3 етапи: «1. Формування обмежень», «2. Підготовка вхідних даних на основі вводу від користувачів», «3. Формування розкладу». Третій етап складається з двох під-етапів: «3.1. Пріоритизація завдань» та «3.2. Обробка конфліктів».

Розроблений комбінований метод на кожному етапі використовує сильні сторони алгоритмів, з яких він складається, компенсуючи їх недоліки та слабкі сторони. А саме: алгоритму на основі обмежень, генетичному алгоритмі, жадібному алгоритмі та моделі ШНМ.

По-перше, було розглянуто використання штучної нейронної мережі для розробки комбінованого методу підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання. Розроблена ШНМ використовується на двох етапах комбінованого методу: «2. Підготовка вхідних даних на основі вводу від користувачів» та «3.2. Обробка конфліктів». ШНМ, за допомогою доповнення вхідних даних від користувачів, даними наближеними до реальних, робить можливим створення оптимального компромісного розкладу занять з урахуванням побажань як від студентів, так і від викладачів курсів.

По-друге, було розглянуто алгоритм на основі обмежень для розробки комбінованого методу підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання.

Алгоритм на основі обмежень є частиною першого етапу методу, а його результати використовується на всіх етапах.

Вихідними даними цього алгоритму є:

- список обмежень;
- список критеріїв значущості та їхніх ваг;
- ієрархічна структура (крім останнього рівня);
- умови оцінювання валідності проміжків часу;
- умови оцінювання завершення ітерації;
- оцінювання успішного створення розкладу.

По третє, було розглянуто використання генетичного алгоритму для розробки комбінованого методу підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання.

Результат генетичного алгоритму – це згруповані проміжки вільного часу користувачів, з вказанням кількості співпадінь вподобань. Список проміжків упорядковано за критерієм значущості «Кількість співпадінь» в низхідному порядку.

Генетичний алгоритм є частиною другого етапу комбінованого методу.

В четвертих, було розглянуто використання жадібного алгоритму для розробки комбінованого методу підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання. Жадібний вибір відбувається на основі результатів інших алгоритмів: алгоритму на основі обмежень, генетичному алгоритмі та ШНМ. Жадібний алгоритм використовується третім етапом методу «3. Формування розкладу» та його двома під-етапами «3.1. Пріоритизація занять», «3.2. Обробка конфліктів», його результатом роботи є список занять, тобто розклад.

Отже, в розділі було розроблено комбінований метод, описано особливості використання всіх його складових частин – алгоритмів. Розгляд цих частин проводився з прив'язкою к трьом етапам методу.

### **3 ДОСЛІДЖЕННЯ ОСОБЛИВОСТЕЙ РЕАЛІЗАЦІЇ КОМБІНОВАНОГО МЕТОДУ І МОДЕЛІ ПІДВИЩЕННЯ ГНУЧКОСТІ РОЗКЛАДУ ІТ-ПРОЄКТУ ДИСТАНЦІЙНОГО НАВЧАННЯ**

3.1 Сценарій використання комбінованого методу і моделі підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання

Ґрунтуючись на описі комбінованого методу підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання та етапах його створення, а також, описі використання алгоритмів та моделі для розробки комбінованого методу: штучної нейронної мережі, алгоритму на основі обмежень, генетичного алгоритму, жадібного алгоритму, було проведено дослідження особливостей реалізації комбінованого методу і моделі [21].

Перший етап комбінованого методу – це етап «1. Формування обмежень».

На цьому етапі, по-перше, необхідно проаналізувати особливості процесу складання розкладу конкретного ІТ-проєкту дистанційного навчання та виділити обмеження розкладу.

Обмеження розкладу – це характеристики розкладу, що впливають на його створення.

Обмеження розкладу відноситься до будь-якого обмеження, яке впливає на час або послідовність завдань або дій у розкладі. Ці обмеження можуть бути накладені зовнішніми факторами, такими як кінцеві терміни, наявність ресурсів або договірні угоди, або вони можуть бути притаманними характеру роботи, що виконується.

Деякі поширені приклади обмежень розкладу включають:

— обмеження ресурсів: це стосується обмежень, накладених наявністю персоналу, обладнання або інших ресурсів, необхідних для виконання завдання;

- часові обмеження: це стосується обмежень, накладених зовнішніми кінцевими термінами або часовими рамками;
- залежності: це відноситься до обмежень, накладених тим фактом, що певні завдання мають бути завершені, перш ніж інші можуть початися.

Результати аналізу обмежень розкладу повинні бути представлені, згідно з ЗВО, у вигляді таблиці з наступним вмістом: множиною змінних –  $X$ , областю значень –  $D$  та множиною обмежень –  $C$  (рис. 3.1).

<b>X – множина змінних</b>	<b>D – область значень</b>	<b>C – множина обмежень</b>
↓ <b>Перелік обмежень</b>	↓ <b>Нижній та верхній поріг значення обмеження / його константне вираження</b>	↓ <b>Перелік допустимих значень</b>
<b>Формат подання</b>		
<b>обмеження (назва)</b>	[мінімум; максимум] чи у вигляді нерівності / константне значення	{значення 1, значення 2} за необхідності: одиниці вимірювання та коментар
<b>Приклад</b>		
<b>тривалість уроку</b>	15 хвилин (мінімальна тривалість) ≤ тривалість уроку ≤ 120 хвилин (максимальна тривалість)	{15, 45, 60, 120} хвилин в залежності від налаштувань курсу

Рисунок 3.1 – Приклад формату таблиці результатів аналізу обмежень розкладу

З загального списку обмежень потрібно виділити ті, з яких сформулюються критерії значущості та ваги.

Тобто, по-друге, треба провести аналіз критеріїв значущості та їхніх ваг.

Результат аналізу критеріїв значущості представляє собою ієрархічну структуру, упорядковану за значеннями ваг.

Перший рівень структури – кореневий, на цей рівень ставиться найбільш загальна сутність, тобто така з якою пов’язані всі інші. Її список також може бути упорядкованим, а значення її ваги буде завжди найбільшим.

Наступні рівні ієрархії обумовлюються специфікою конкретного ІТ-проєкту і можуть бути пов’язані наприклад з акціями, спеціальними акаунтами, тощо.

Останній рівень ієрархії – найважливіший, він представляє собою складові частини розкладу. Цей рівень складається з проміжків часу користувачів перетворених і злитих генетичним алгоритмом. При цьому його характеристики можуть мати різні ваги й складні правила сортування відповідно.

Результати аналізу критеріїв значущості та їхніх ваг подаються у вигляді таблиці з наступним вмістом: тип сутності, назва параметру, критерій значущості, вага (рис. 3.2).

Тип сутності	Назва параметру	Критерій значущості	Вага	
↓ Перелік сутностей	↓ Перелік властивостей сутностей	↓ Значення до якої прагне властивість (тобто чим ближче значення до граничного, тим більший пріоритет має сутність)	↓ Числове вираження пріоритету сутності (чим вище – тим вищий рівень в ієрархії)	
<b>Формат подання</b>				
сутність (назва)	параметр (назва)	параметр → граничне значення	вага параметру (якщо у сутності їх > 1)	натуральне число від 1

Рисунок 3.2 – Таблиця результатів аналізу критеріїв значущості, їхніх ваг

Наступним етап йде формування умов: умови оцінювання завершення ітерації, умови оцінювання успішного створення розкладу.

Умови оцінювання завершеності ітерації залежать від локального створення розкладу для кореневої сутності. Вони цілком залежать від особливостей проєкту, в загальному випадку – чи відповідає створена кількість занять бажаній. Можливі модифікації, наприклад, створення тільки в ті проміжки де процент дорівнює бажаному.

Умови оцінювання успішного створення розкладу – можуть складатися з локальних успіхів створення на ітераціях, у базовому варіанті.

Останній рівень ієрархічної структури, правила його створення – вхідні дані для другого етапу «Підготовка вхідних даних на основі вводу від користувачів».

Етап «Підготовка вхідних даних на основі вводу від користувачів» займається перетворенням останнього рівня ієрархічної структури.

По-перше, формується вигляд базової сутності проміжку, в тому вигляді, в якому він зберігається в системі.

Обов'язково повинні бути параметри початку і кінця проміжку у вигляді дата-час. Додаткові поля формуються виходячи з результатів аналізу обмежень системи.

По-друге, формується кінцевий варіант сутності, він повинен містити всі ті поля, що вказані в аналізі критеріїв ефективності.

По-третє, за вказаним в методі алгоритмом документуються перетворення проміжків. В базовому методі вказані такі віхи за поколіннями:

- перше покоління: «1. Проміжки вільного часу», тобто формування базової сутності проміжку;
- друге покоління: «2.1. Створення додаткових проміжків ШНМ» та «2.2. Фільтрація на основі проміжків викладачів»;
- третє покоління: «3.1. Злиття проміжків з вказанням кількості співпадінь», «3.2. Фільтрація співпадінь з вже створеними уроками».

За необхідності, усі пункти крім першого можуть бути модифіковані за умови виконання обов'язкових вимог до кожного з них.

Віха «2.1. Створення додаткових проміжків ШНМ» має наступні вимоги:

- створювати проміжки тільки якщо, користувач задав не достатню кількість слотів (перевірка на недостатню кількість слотів може бути адаптованою під окремі проєкти);
- вхідні дані до ШНМ повинні бути подані як базова сутність проміжку.

Віха «2.2. Фільтрація на основі проміжків викладачів» в якості вхідних даних базові сутності проміжку, зі сховища даних та згенеровані ШНМ:

- відфільтрувати дані так, щоб проміжки студентів та викладача становили однорідну сутність;
- за необхідністю розробити інший формат для цього покоління.

Віха «3.1. Злиття проміжків з вказанням кількості співпадінь» – головне перетворення базових проміжків на проміжки ієрархічної структури:

- перетворити згідно з критеріями значущості, за допомогою алгоритму злиття проміжків, з базових до цільових;
- вказати кількість батьківських з яких створили цільовий.

Віха «3.2. Фільтрація співпадінь з вже створеними уроками» – крок для відкидання проміжків, що мають перетин з існуючими заняттями. Можна використовувати базовий алгоритм, чи ускладнити його в залежності від обмежень проєкту.

Етап «3. Формування розкладу» – останній етап, створення розкладу жадібним вибором.

Під-етап «3.1. Пріоритизація завдань» – реалізація проходу по ієрархічній структурі з використанням розроблених умов завершення ітерації.

Під-етап «3.2. Обробка конфліктів» – реалізує створення заняття, та

оброблює конфлікти, що виникають – тобто не достатність проміжків.

Отже, створення розкладу за комбінованим методом проходить в три етапи, кожен з яких має свої обов'язкові частини.

Кожен з етапів також включає в себе частини-віхи, що можна адаптувати під особливості проекту.

Результати етапів:

- таблиця результатів аналізу обмежень розкладу;
- таблиця результатів аналізу критеріїв значущості, їхніх ваг;
- формально визначені умови: оцінювання завершення ітерації, умови оцінювання успішного створення розкладу;
- опис базової сутності проміжку розкладу;
- опис кінцевого варіанту сутності проміжку, як частини ієрархічної структури;
- алгоритм перетворень проміжків, з вказанням перетворень за трьома поколіннями та їх віхами, а саме: «1. Проміжки вільного часу», «2.1. Створення додаткових проміжків ШНМ», «2.2. Фільтрація на основі проміжків викладачів», «3.1. Злиття проміжків з вказанням кількості співпадінь», «3.2. Фільтрація співпадінь з вже створеними уроками»;
- ієрархічна структура;
- опис алгоритму під-етапу «3.1. Пріоритизація завдань»: проходження по структурі, використання умови завершення ітерації;
- опис алгоритму під-етапу «3.2. Обробка конфліктів»: створення занять, вирішення конфлікту нестачі проміжків, використання умови оцінювання успішного створення розкладу.

Отримані результати – готовий матеріал для автоматизації створення гнучкого розкладу занять. При цьому, реалізація є платформо незалежною та може складатися з кількох модулів, а самі результати будуть різнитися в залежності від конкретного проекту, його обмежень та умов, процесів функціонування.

### 3.2 Опис взаємодії користувачів з комбінованим методом за допомогою User-story

Так як викладач та студент – це ті користувачі, що безпосередньо користуються ГРЗ у системі та впливають на його створення, то розглядання їх функціоналу у вигляді User-story, допоможе зрозуміти процес взаємодії комбінованого методу та кінцевих користувачів.

User-story (історія користувача) – це невелика самодостатня одиниця розробки, призначена для досягнення певної мети в рамках продукту. User-story зазвичай пишеться з точки зору певної особи-ролі користувача та має такий формат: «Як [персона користувача], я хочу [виконати цю дію], щоб [я міг досягти цієї мети]» [22].

Acceptance Criteria (AC), з англійської – критерії прийняття. Тобто, AC – це набір конкретних вимог, що піддаються вимірюванню, яким має відповідати продукт або функція, щоб бути прийнятими клієнтом або користувачем. Правильно написані User-story повинні відповідати стандартам INVEST (independent, negotiable, valuable, estimable, small, and testable), стисле пояснення (рис. 3.3) [22].

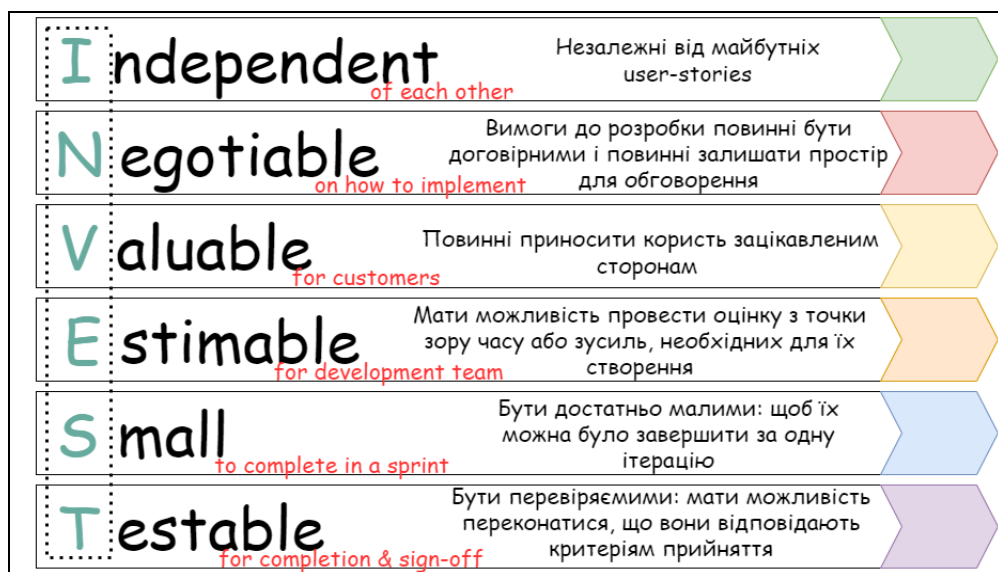


Рисунок 3.3 – Розшифровка INVEST з стислим описом

Погляд зі сторони викладача охоплює роботу з заняттями розкладу, студентами, курсом, навчальної програмою, тощо.

User-story з точки зору викладача дистанційних курсів на функціональність пов'язану з створенням та взаємодією з ГРЗ, розробленим за допомогою комбінованого методу підвищення гнучкості розкладу ІТ-проекту дистанційного навчання, подано у таблиці 3.1.

Таблиця 3.1 – User-story з точки зору викладача дистанційних курсів на функціональність пов'язану з створенням та взаємодією з ГРЗ

User-story (історія користувача)	Acceptance Criteria (критерії прийняття)
Як <b>Викладач, я хочу, щоб</b> розклад враховував проміжки мого вільного часу, <b>щоб</b> я міг користуватися перевагами віддаленої праці і поєднувати викладацьку діяльність з моїми власними справами чи іншою роботою	інтерфейс викладача повинен мати календар, де б можна було зазначати (на майбутні періоди) та переглядати проміжки вільного часу
	створення розкладу повинно враховувати проміжки вільного часу викладача у пошуку компромісного часу проведення занять
	викладачі мають бути проінформовані о термінах внесення інформації о проміжках
Як <b>Викладач, я хочу, мати</b> можливість переглядати та редагувати тему та опис майбутнього заняття своїх курсів, <b>щоб</b> я міг переконатися, що студенти мають змогу ознайомитися з навчальними цілями	інтерфейс викладача повинен надавати змогу відредагувати тему та опис майбутніх заняття
	інтерфейс викладача повинен надавати можливість переглядати тему та опис занять своїх курсів
Як <b>Викладач, я хочу, щоб</b> розклад включав поєднання синхронних і асинхронних навчальних дій, а саме – щоб заняття мали посилання на відеоконференцію та на матеріали на самостійне опрацювання, <b>щоб</b> студенти могли працювати з матеріалом різними способами	інтерфейс викладача повинен надавати можливість переглядати посилання на відеоконференцію заняття та матеріали на самостійне опрацювання для занять своїх курсів
	інтерфейс викладача повинен надавати можливість редагувати посилання на відеоконференцію заняття та на матеріали до самостійного опрацювання для майбутніх занять своїх курсів

Кінець таблиці 3.1

User-story (історія користувача)	Acceptance Criteria (критерії прийняття)
Як <b>Викладач, я хочу, щоб</b> у розкладі було достатньо часу для студентів, <b>щоб</b> виконати завдання та оцінити, <b>щоб</b> вони мали змістовний досвід навчання	в системі повинно бути налаштування проміжку часу між заняттями одного курсу
	розклад повинен враховувати налаштування проміжку часу між заняттями одного курсу
Як <b>Викладач, я хочу переконатися, що</b> розклад містить відповідні перерви та простої, <b>щоб</b> я міг мати час на відпочинок та підготовку до наступного заняття за розкладом	в системі повинно бути налаштування довжини перерви між заняттями
	розклад повинен враховувати налаштування довжини перерви й не ставити інші заняття перед та після курсу в цей проміжок
Як <b>Викладач, я хочу мати змогу</b> переглядати розклад у вигляді календарю з подіями-заняттями на майбутні та попередні періоди, <b>щоб</b> я міг планувати свій час та освіжити у пам'яті минулі заняття	інтерфейс викладача повинен мати календар, де було б можливо переглядати заняття
	інтерфейс викладача повинен відображати до якого курсу належить заняття
Як <b>Викладач, я хочу, щоб</b> у разі відсутності заданих проміжків вільного часу чи їх недостатньої кількості створення розкладу аналізувало й враховувало данні за попередні періоди, <b>щоб</b> мої часові вподобання враховувалися навіть у разі відсутності даних від мене	створення розкладу повинно вміти аналізувати та доповнювати данні о проміжках вільного часу викладача у разі відсутності заданих проміжків вільного часу чи їх недостатньої кількості
	створення розкладу повинно враховувати спрогнозовані дані проміжків вільного часу викладача

User-story з точки зору студента дистанційних курсів на функціональність пов'язану з створенням та взаємодією з ГРЗ, розробленим за допомогою комбінованого методу підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання, подано у таблиці 3.2.

Таблиця 3.2 – User-story з точки зору студента дистанційних курсів

User-story (історія користувача)	Acceptance Criteria (критерії прийняття)
Як Студент, я хочу, щоб розклад враховував проміжки мого вільного часу, щоб я міг поєднувати навчання з іншою меєю діяльністю, власними справами	інтерфейс студента повинен мати календар, де б можна було зазначати (на майбутні періоди) та переглядати проміжки вільного часу
	створення розкладу повинно враховувати проміжки вільного часу студента у пошуку компромісного часу проведення занять
	студенти мають бути проінформовані о термінах внесення інформації о проміжках
Як Студент, я хочу мати змогу переглядати розклад у вигляді календарю з подіями-заняттями на майбутні та попередні періоди, щоб я міг планувати свій час та освіжити у пам'яті минулі заняття	інтерфейс студента повинен мати календар, де було б можливо переглядати заняття
	інтерфейс студента повинен відображати до якого курсу належить заняття
Як Студент, я хочу, мати можливість переглядати тему та опис майбутнього заняття своїх курсів, щоб я міг ознайомитися та підготуватися до занять	інтерфейс студента повинен надавати змогу переглядати тему та опис майбутніх занять своїх курсів
Як Студент, я хочу мати можливість переглядати посилання на відеоконференцію та на матеріали для самостійного опрацювання, щоб я міг виконувати завдання та мати змогу взаємодіяти з викладачем онлайн	інтерфейс студента повинен надавати можливість переглядати посилання на відеоконференцію заняття та на матеріали до самостійного опрацювання для занять своїх курсів
Як Студент, я хочу, щоб, у разі відсутності заданих проміжків вільного часу чи їхній нестачі, створення розкладу аналізувало й враховувало данні за попередні періоди, щоб розклад все одно адаптувався під мої потреби	створення розкладу повинно вміти аналізувати та доповнювати проміжки вільного часу студента у разі їх відсутності чи нестачі
	створення розкладу повинно враховувати спрогнозовані дані проміжків вільного часу студента

Отже, виходячи з User-story викладача та студента можна зробити висновок об необхідних елементах користувацького інтерфейсу. А також о системних налаштуваннях-обмеженнях, що впливають на викладацьку діяльність напрямку.

Перелік налаштування системи пов'язані з діяльністю викладача:

- довжина перерви;
- проміжок часу між заняттями одного курсу (за потреби).

Перелік необхідних елементів користувацького інтерфейсу викладача:

- календар проміжків вільного часу з можливістю редагування майбутніх періодів;
- календар занять курсів;
- пов'язану, із заняттями з календарю, форму для редагування заняття, яка містить в собі наступні поля: «Тема заняття», «Опис заняття», «Посилання на відеоконференцію», «Посилання на матеріали для самостійного опрацювання».

Перелік необхідних елементів користувацького інтерфейсу студента:

- календар проміжків вільного часу з можливістю редагування майбутніх періодів;
- календар занять курсів;
- пов'язаний, із заняттями з календарю інформаційний елемент будь-якого типу з обов'язковим відображенням таких полів: «Тема заняття», «Опис заняття», «Посилання на відеоконференцію», «Посилання на матеріали для самостійного опрацювання».

Створені User-story дають загальне представлення проходження взаємодії користувачів – студентів та викладачів, з комбінованим методом. Також було наведено пояснення, з точки зору користувачів, необхідності використання ШНМ задля генерації наближених до реальних проміжків вільного часу.

На основі аналізу взаємодії користувачів та комбінованого методу була створена Use-case діаграма. Use-case, з англійської, діаграма варіантів використання – це тип діаграми UML (Unified Modeling Language), яка представляє взаємодію між актором (користувачем або системою) і системою. Вона надає візуальне представлення різних способів взаємодії користувача з системою, а також різних функцій, які система надає користувачеві [23].

Use-case діаграма взаємодії викладача та студента з комбінованим методом надає візуальне представлення взаємодії користувачів, елементів інтерфейсу та функцій системи (рис. 3.4).

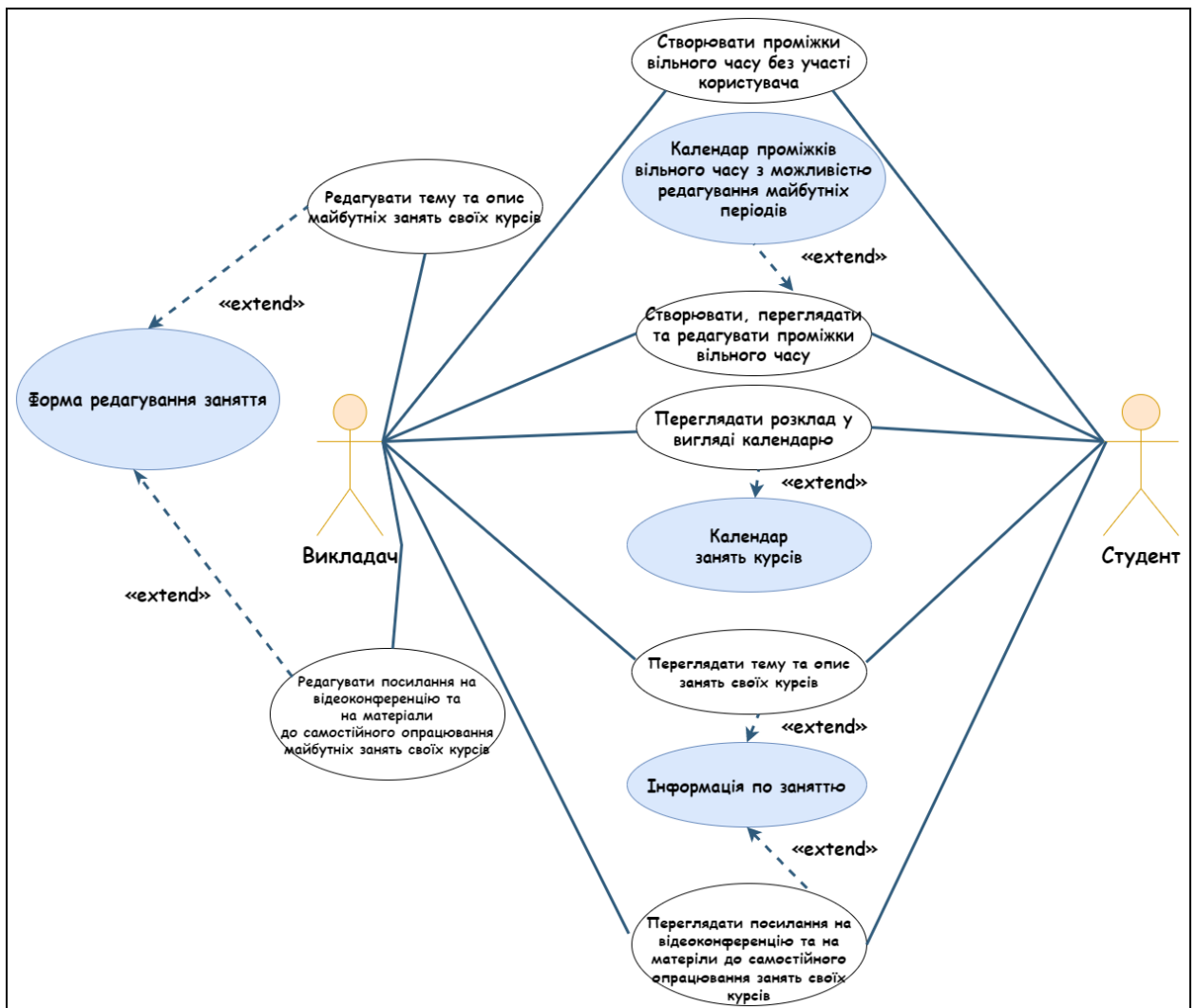


Рисунок 3.4 – Use-case діаграма взаємодії викладача та студента з комбінованим методом

### 3.3 Опис реалізації комбінованого методу на платформно-незалежному рівні

Автоматизація комбінованого методу створення розкладу, back-end частини, вимагає використання наступних складових:

- time-trigger функція, що відпрацьовує за Cron-виразом;
- клас складання розкладу (ScheduleService) та клас з реалізацією логіки (IScheduleCreator) створення занять (EagerScheduleCreator);
- клас ієрархічної структури (ScheduleHierarchy);
- рівень взаємодії з даними, Data Access Layer (DAL);
- модуль ШНМ (ANNModule);
- сховище даних (Data storage).

Взаємодія між цими складовими подана на рис 3.5.

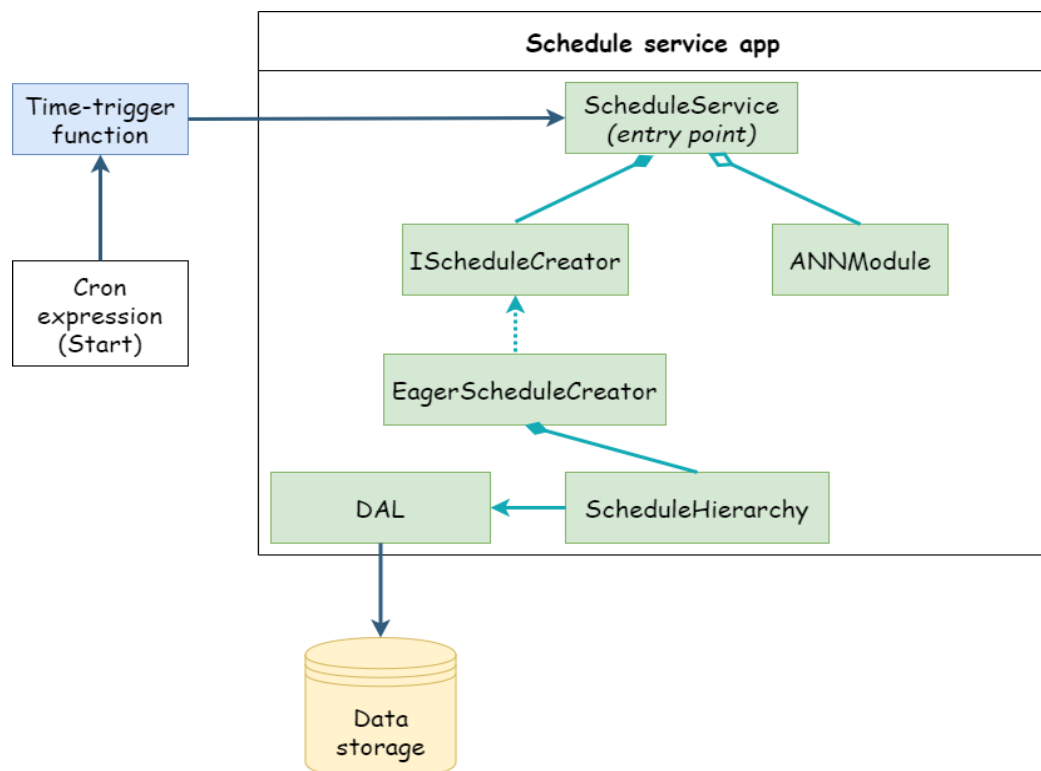


Рисунок 3.5 – Взаємодія між складовими реалізації комбінованого методу

Так як створення розкладу – це рутинна повторювана справа, що відбувається наступним чином: користувачі системи періодично заповнюють свої проміжки вільного часу, на основі них розклад складається в час найменшої активності. При цьому створення розкладу повинно відбуватися за вказаний проміжок часу до початку періоду, на який треба скласти розклад. Тобто, наприклад, розклад складається раз на тиждень, введені проміжки на наступний період фіксуються в п’ятницю і о 00:00 за тайм-зоною тригериться функція створення розкладу. Зазвичай такі правила запуску прописуються у вигляді Cron-виразу, приклад формату якого подано на рис. 3.6.

## “At 00:00 on Friday.”

next at 2023-03-17 00:00:00 random

0
00
\*
\*
5

<u>minute</u>	<u>hour</u>	<u>day</u> (month)	<u>month</u>	<u>day</u> (week)
<hr/>				
		*	any value	
		,	value list separator	
		-	range of values	
		/	step values	
		@yearly	(non-standard)	
		@annually	(non-standard)	
		@monthly	(non-standard)	
		@weekly	(non-standard)	
		@daily	(non-standard)	
		@hourly	(non-standard)	
		@reboot	(non-standard)	

Рисунок 3.6 – Приклад Cron-виразу

Тайм функція може бути у вигляді бекграунд-процесу чи використовувати без серверне (serverless) рішення, таке як AWS Lambda чи Azure Functions.

Сама по собі функція зазвичай не містить логіки й викликає метод сервісу ScheduleService, що в свою чергу містить клас створення розкладу з прописаною у ньому логікою і власними залежностями. Для з ієрархічною структурою відповідаю клас ScheduleHierarchy. Зв'язком з сховищем даних займається рівень доступу до даних – DAL. ANNModule – містить функціонал по роботі з ШНМ відповідно.

Наведене розподілення відповідає SOLID, в тому числі принципу єдиної відповідальності Single Responsibility Principle. SOLID – аббревіатури п'яти принципів дизайну об'єктно-орієнтованого програмування. Ці принципи спрямовані на те, щоб зробити проєкти програмного забезпечення більш зрозумілими, гнучкими та придатними для обслуговування з часом. Single Responsibility Principle стверджує, що клас або модуль повинен мати лише одну причину для зміни. Це означає, що клас повинен мати лише одну відповідальність або мету, і що він не повинен нести відповідальності більше ніж за одну річ. Цей принцип сприяє поділу проблем і робить код більш модульним, придатним для тестування та обслуговуванням [24].

DAL – це архітектурний шаблон програмного забезпечення, який відокремлює логіку доступу до даних від інших компонентів програми. DAL забезпечує стандартизований спосіб для різних частин програми доступу до даних з одного або кількох джерел даних, таких як база даних, веб-служба або файлова система [24].

DAL зазвичай складається з набору класів або модулів, які визначають інтерфейси та методи доступу до даних. Він інкапсулює деталі того, як дані зберігаються та витягуються, і забезпечує рівень абстракції між програмою та основними джерелами даних. Такий розподіл завдань полегшує підтримку та зміну програми з часом [25].

Крім надання рівня абстракції, DAL також може надати інші переваги, такі як покращена безпека, краща продуктивність і легше тестування. Завдяки централізації логіки доступу до даних на окремому рівні стає легше захищати та перевіряти доступ до конфіденційних даних. DAL також може оптимізувати операції доступу до даних для підвищення продуктивності та надати зручний спосіб імітації або заглушки джерел даних для цілей тестування [25].

Загалом, DAL є важливою моделлю в архітектурі програмного забезпечення, яка допомагає просувати модульність, зручність обслуговування та масштабованість сучасних програм.

Отже, вище наведені складові відповідають сучасним методам розробки, а саме архітектурному шаблону DAL та принципам SOLID. Поданий сценарій взаємодії дає представлення о роботі з сервісом складання розкладу. Створені заняття інші сервіси можуть зчитувати з сховища даних чи отримувати з Schedule Service app за допомогою HTTP-запиту рис 3.7.

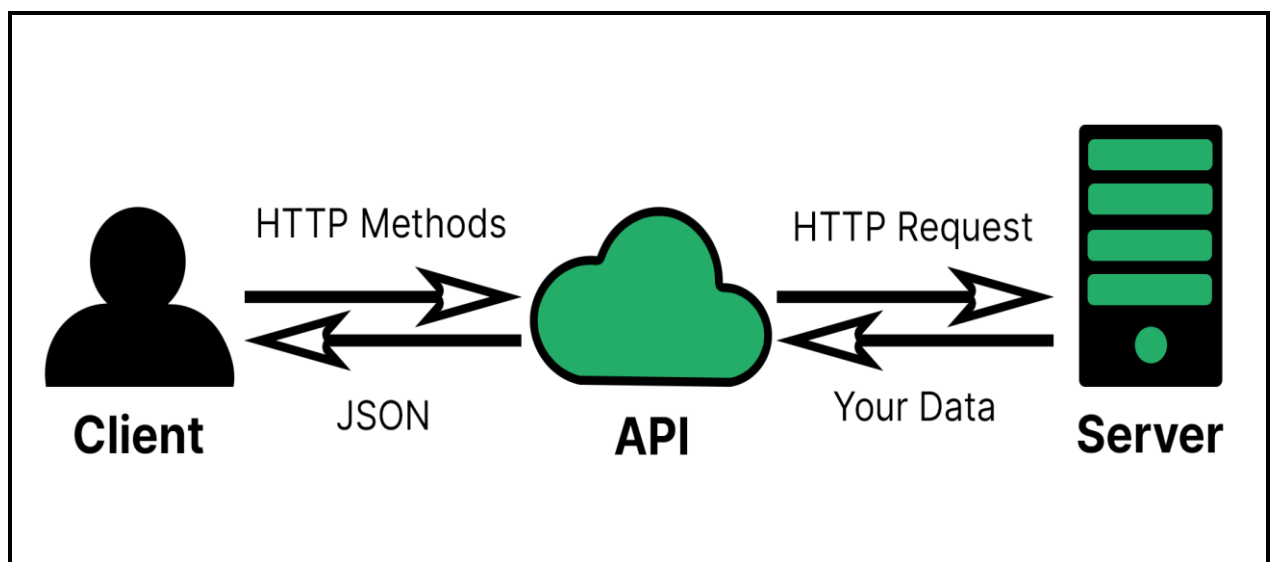


Рисунок 3.7 – Приклад взаємодії з Schedule Service app

У якості клієнта може виступати інший back-end додаток чи сервіс, front-end додаток, функція тощо.

### 3.4 Висновки з дослідження особливостей реалізації комбінованого методу підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання

За результатами розробки комбінованого методу підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання, а саме: опису комбінованого методу підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання, та етапів створення розкладу; використання штучної нейронної мережі для розробки комбінованого методу підвищення гнучкості розкладу; використання алгоритму на основі обмежень для розробки комбінованого методу підвищення гнучкості, використання генетичного алгоритму для розробки комбінованого методу підвищення гнучкості розкладу, використання жадібного алгоритму для розробки комбінованого методу підвищення гнучкості розкладу – було проведено дослідження особливостей реалізації комбінованого методу підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання.

По перше, на основі дослідження сценарію використання комбінованого методу і моделі підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання, було отримано список результатів кожного з трьох етапів створення розкладу, з описом форматів їх подання.

По друге, було проаналізована та описано взаємодію користувачів з комбінованим методом за допомогою User-story. User-story були розроблені з дотриманням стандартів INVEST.

Проведений опис взаємодії дозволив виділити необхідні елементи користувацького інтерфейсу студентів та викладачів, створити перелік налаштування системи пов'язані з діяльністю викладача. Крім того, було визначено пояснення необхідності, з точки зору користувачів методу, використання ШНМ задля генерації наближених до реальних проміжків вільного часу.

Також, на основі User-story з точки зору викладача та студента було створено Use-case діаграму взаємодії викладача та студента з комбінованим методом з метою надання візуального представлення взаємодії користувачів, елементів інтерфейсу та функцій системи.

По-третє, проведений опис реалізації комбінованого методу на платформно-незалежному рівні дозволив дослідити на верхньому рівні взаємодію між складовими реалізації комбінованого методу. У якості складових back-end частини виступають:

- time-trigger функція, що відпрацьовує за Cron виразом;
- клас складання розкладу (ScheduleService) та клас з реалізацією логіки (IScheduleCreator) створення занять (EagerScheduleCreator);
- клас ієрархічної структури (ScheduleHierarchy);
- рівень взаємодії з даними, Data Access Layer (DAL);
- модуль ШНМ (ANNModule);
- сховище даних (Data storage).

Було підкреслено відповідність такого поділення сучасним методам розробки, а саме архітектурному шаблону DAL та принципам SOLID.

Отже, в розділі було проведено дослідження особливостей реалізації комбінованого методу підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання. За результатами якого було отримано представлення про сценарій використання методу, процес взаємодії користувачів з методом, та високорівневе представлення складових back-end частини і їх взаємозв'язок.

## 4 АПРОБАЦІЯ КОМБІНОВАНОГО МЕТОДУ ТА МОДЕЛІ ПІДВИЩЕННЯ ГНУЧКОСТІ РОЗКЛАДУ ІТ-ПРОЄКТУ ДИСТАНЦІЙНОГО НАВЧАННЯ

4.1 Аналіз предметної галузі, що визначається діяльністю фірми дистанційних курсів

За завданням кваліфікаційної роботи розглядається діяльність фірми, що надає послуги з навчання на дистанційних курсах. Фірма надає послуги за грошову винагороду яка встановлюється для кожного курсу окремо. В роботі фірми, що розглядається, задіяні такі працівники як вчителі та адміністратори.

Організаційна структура підприємства, що розглядається подана на рис. 4.1.

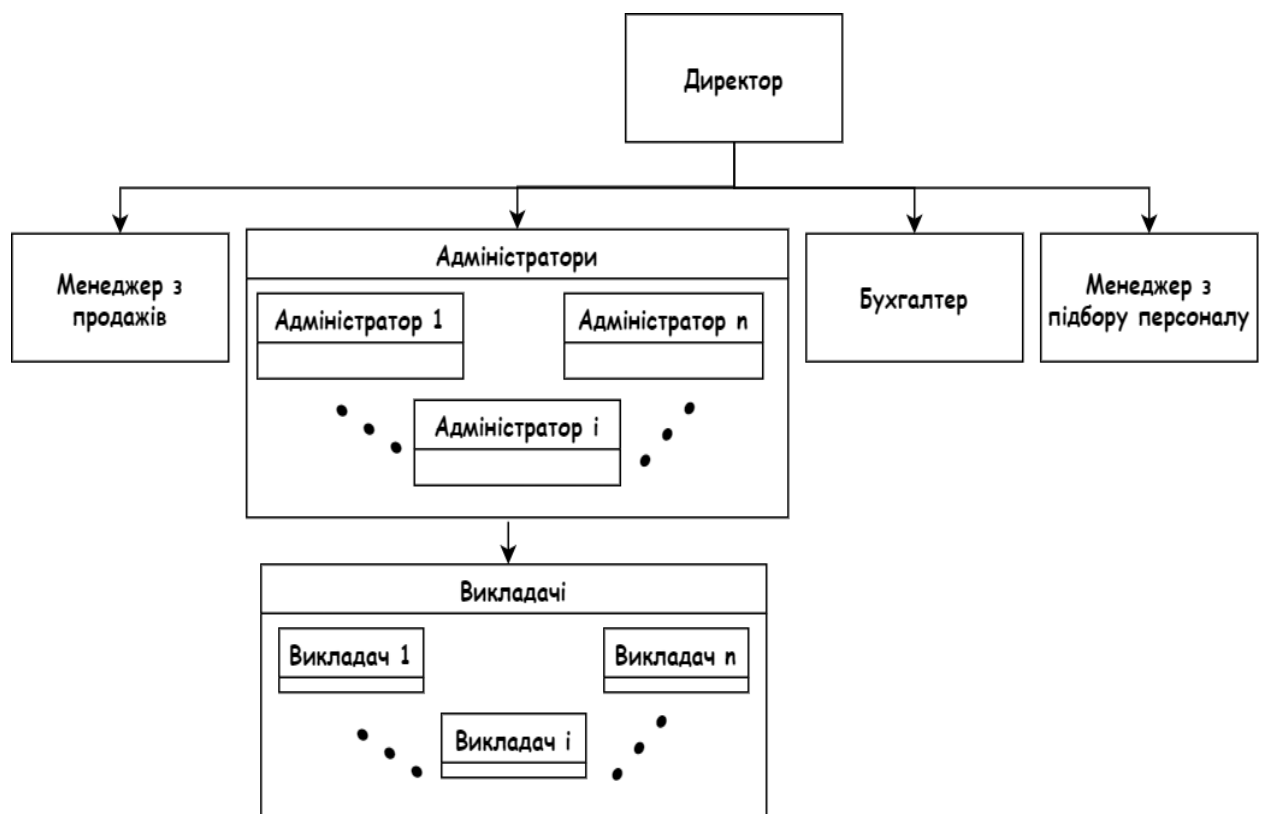


Рисунок 4.1 – Схема організаційної структури об'єкту автоматизації

Діяльність фірми дистанційних курсів регулюється такими нормативними актами як:

- господарський кодекс України [26];
- закон України «Про захист інформації в інформаційно-телекомунікаційних системах» [27];
- закон України «Про захист персональних даних» [28].

Директор – одночасно засновник і «обличчя» компанії, йому безпосередньо підпорядковуються менеджер з продажів, менеджер з підбору персоналу, бухгалтер та штат адміністраторів.

Менеджер з продажів відповідає за розміщення реклами підприємства, просуванню його на ринку, збільшенню кількості студентів.

Менеджер з підбору персоналу спеціалізується на пошуку та перевірці кандидатів на посади адміністраторів та викладачів.

В якості бухгалтера виступає сторонній, позаштатний бухгалтер, що організовує та веде бухгалтерський облік фірми.

Адміністраторів в фірмі декілька, вони працюють по-змінно. Кожному адміністратору підпорядковуються вчителі зміни. Тобто, адміністратори не прив'язані до конкретних викладачів.

В обов'язки адміністратору входить:

- облік викладачів;
- керування та взаємодія з викладачами;
- облік курсів;
- облік студентів;
- взаємодія з студентами у якості технічної підтримки;
- облік та супроводження замовлень запису на курс (під супроводженням мається на увазі підтвердження запису, видача розкладу, прийняття оплати та відповіді на організаційні питання студента).

Викладачів, як і адміністраторів, в фірмі декілька, за кожним викладачем закріплено певні курси фірми. Теми, наповнення курсів – входять до обов'язків викладача.

Крім того, до викладача висуваються спеціальні вимоги – досвід викладання не менше одного року та документ, що підтверджує його кваліфікацію в галузі, в рамках якої він планує проводити дистанційні заняття.

Загальними особливостями роботи фірми є:

- студент може проходити лише один курс одночасно, для підвищення ефективності занять за рахунок зосередженості на одному напрямку;
- в день проводиться одне заняття для кожного курсу;
- кількість занять на тиждень варіюється від курсу к курсу і встановлюється до першого заняття і до закінчення залишається незмінною;
- курс веде один викладач задля формування найліпшого підходу до студентів-слухачів.

Типовий клієнт фірми – це людина від 18 до 50 років, що навчається або працює і не має змоги відвідувати очні заняття. При цьому на курс записують невеликі групи людей, що є значно дешевшим ніж індивідуальні заняття, а значить й доступнішим.

В середньому послугами фірми одночасно користується приблизно менше ста студентів, в штаті працює до двадцяти викладачів та йде не більше 10 курсів.

Отже, організаційна структура підприємства включає такі ланки: директор, менеджер з продажів, менеджер з підбору персоналу, бухгалтер, адміністратори, викладачі. Було визначено типового клієнта фірми, загальні особливості її роботи.

## 4.2 Визначення сфери застосування комбінованого методу підвищення гнучкості розкладу ІТ-проєкту для фірми дистанційних курсів

Для визначення сфери застосування розроблювальної системи розглядався бізнес-процес створення розкладу, що реалізовано у фірмі в поточний час.

Адміністратори фірми відповідають за складання розкладу на тиждень у стандартній системі «день тижня» «час» – «заняття» для кожного курсу. При цьому мають виконуватися умови: лише одне заняття курсу в день, навантаження викладача, наявність перерв.

Тобто, бізнес-функція створення розкладу занять реалізується адміністратором і подається у вигляді таблиць офісного пакету Excel, створення розкладу відбувається фіксовано.

Розклад створюються для курсу з моменту його старту й залишається незмінним до його кінця. Тобто, курс з трьома заняттями в неділю завжди буде проходити по одній схемі «Понеділок, Середа, П'ятниця».

Базову інформацію з курсів надають викладачі, ці дані також вносяться до таблиці розкладу. Під базовою інформацією мається на увазі: назва заняття та домашнє завдання у вигляді посилання на електронний документ.

За політикою фірми, одночасно студент може бути слухачем лише одного курсу.

Мета використання комбінованого методу – це залучення нових клієнтів-студентів та вчителів, за рахунок створення розкладу автоматично та гнучким способом.

Отже, має бути автоматизовано бізнес-процес фірми – складання розкладу на основі побажань студентів та викладачів з використанням комбінованого методу підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання.

### 4.3 Аналіз використання комбінованого методу підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання у порівнянні з фіксованим графіком

Задля доведення ефективності комбінованого методу підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання було проведено порівняння з існуючим фіксованим розкладом занять (ФРЗ).

Під ФРЗ, мається на увазі той, що зараз реалізовано у фірмі дистанційних курсів – а саме, розклад що призначено на певні дні на певний час з початку і до кінця, наприклад, кожного понеділка і середи о 17:00.

Виходячи з особливостей роботи фірми використання методу підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання буде відбуватися наступним чином: розклад складається щотижня у п'ятницю опівночі, за київським часом, на наступний тиждень.

Заявки на складання розкладу приймаються до початку роботи алгоритму, проміжки додані користувачами після – не враховуються.

#### 4.3.1 Вихідні дані для комбінованого методу, що пов'язані з діяльністю фірми дистанційних курсів

Перелік вихідних даних фірми дистанційних курсів до експерименту з складання розкладу на період, що дорівнює одному тижню:

- дані по трьох курсах одного викладача «Викладач 1»;
- розклад курсів згідно фіксованого графіку;
- проміжки вільного часу викладача на тиждень;

— проміжки вільного часу трьох студентів кожного курсу на тиждень.

Дані по трьох курсах одного викладача «Викладач 1», що будуть застосовуватися в роботі комбінованого методу підвищення гнучкості розкладу фірми дистанційних курсів, подано у таблиці 4.1.

Таблиця 4.1 – Параметри курсів, що є вихідними даними експерименту

Параметри курсу	Курс 1	Курс 2	Курс 3
кількість занять на тиждень	2	3	4
ідентифікатори студентів	1, 2, 3	4, 5, 6	7, 8, 9

Отже, Викладач «Викладач 1» веде три курси «Курс 1», «Курс 2», «Курс 3», кожний курс відвідує по три студенти: «Курс 1» – «Студент 1», «Студент 2», «Студент 3», і т.д. відповідно.

Розклад курсів згідно фіксованого графіку, що встановлено заздалегідь, подано у таблиці 4.2.

Таблиця 4.2 – Розклад курсів згідно фіксованого графіку занять

Параметри розкладу	Курс 1	Курс 2	Курс 3
Дні тижня	субота, неділя	понеділок, середа, п'ятниця	понеділок, вівторок, четвер, п'ятниця
Час проведення занять	12:00-12:45	17:00-17:45	18:00-18:45

Календарне представлення розклад курсів згідно фіксованого графіку з вихідних даних наведено на рис 4.2.

08							
	пн 13	вт 14	ср 15	чт 16	пт 17	сб 18	нд 19
all day							
11							
12						К-1	К-1
13							
14							
15							
16							
17	К-2		К-2		К-2		
18	К-3	К-3		К-3	К-3		
19							
20							
21							
22							
23							

Рисунок 4.2 – Фіксований розклад з вихідних даних курсів викладача «Викладач 1»

Проміжки вільного часу викладача «Викладач 1», що будуть засовуватися у якості обмеження розкладу на тиждень, подано у таблиці 4.3.

Таблиця 4.3 – Проміжки вільного часу викладача на тиждень

День тижня	Проміжки
понеділок	9:00-12:00, 16:00-20:00
вівторок	6:00-10:00, 16:00-18:00
середа	16:00-18:00, 19:00-21:00
четвер	8:00-12:00, 14:00-17:00, 20:00-21:00
п'ятниця	-
субота	9:00-12:00, 19:00-21:00
неділя	8:00-13:00, 20:00-21:00

Проміжки вільного часу студентів на тиждень подано у таблиці 4.4.

Таблиця 4.4 – Проміжки вільного часу студентів на тиждень

№	Проміжки						
	Пн	Вт	Ср	Чт	Пт	Сб	Нед
1	07:00-09:45 13:00-14:00 19:00-20:45	07:00-09:45 12:00-15:00	18:00-20:00	-	-	07:00-10:00	17:00-20:00
2	08:00-10:45 12:00-13:00 18:00-20:00	08:00-10:45 12:00-13:00 18:00-20:00	-	-	-	07:00-14:00	11:00-14:00 19:00-20:00
3	-	07:00-09:45 13:00-14:00 19:00-20:00	-	13:00-20:00	-	09:00-20:00	09:00-20:00
4	06:00-11:00 13:00-15:00 20:00-21:00	-	06:00-11:00 13:00-15:00 20:00-21:00	18:00-21:00	18:00-21:00	13:00-15:00	15:00-18:00
5	06:30-11:30 13:00-14:35 19:00-21:00	-	-	06:30-11:30 13:00-14:35 19:00-21:00	18:00-21:00	07:00-15:00 18:00-20:00	07:00-15:00 18:00-20:00
6	14:00-15:45	07:00-09:45 13:00-14:00 19:00-20:00	07:00-09:45 13:00-14:00 17:00-20:00	13:00-14:00 17:00-20:00	-	07:00-10:00 18:00-21:00	07:00-11:00 19:00-20:00
7	-	05:00-11:00 13:00-14:00	19:00-20:00	05:00-09:45 13:00-14:00 19:00-20:00	05:00-11:00 13:00-14:00	07:00-20:00	-
8	07:00-12:45 16:00-20:00	-	07:00-12:00 13:00-14:00 16:00-20:00	09:00-11:45 16:00-20:00	-	12:00-20:00	17:00-20:00
9	08:00-10:45 12:00-13:00 16:00-18:00 20:00-21:00	-	-	08:00-10:45 13:00-14:00 18:00-20:00	-	07:00-10:00 16:00-21:00	11:00-18:00

Вихідні дані проміжків вільного часу студентів, у календарному вигляді, подано на рис. 4.2, викладача – рис 4.3.

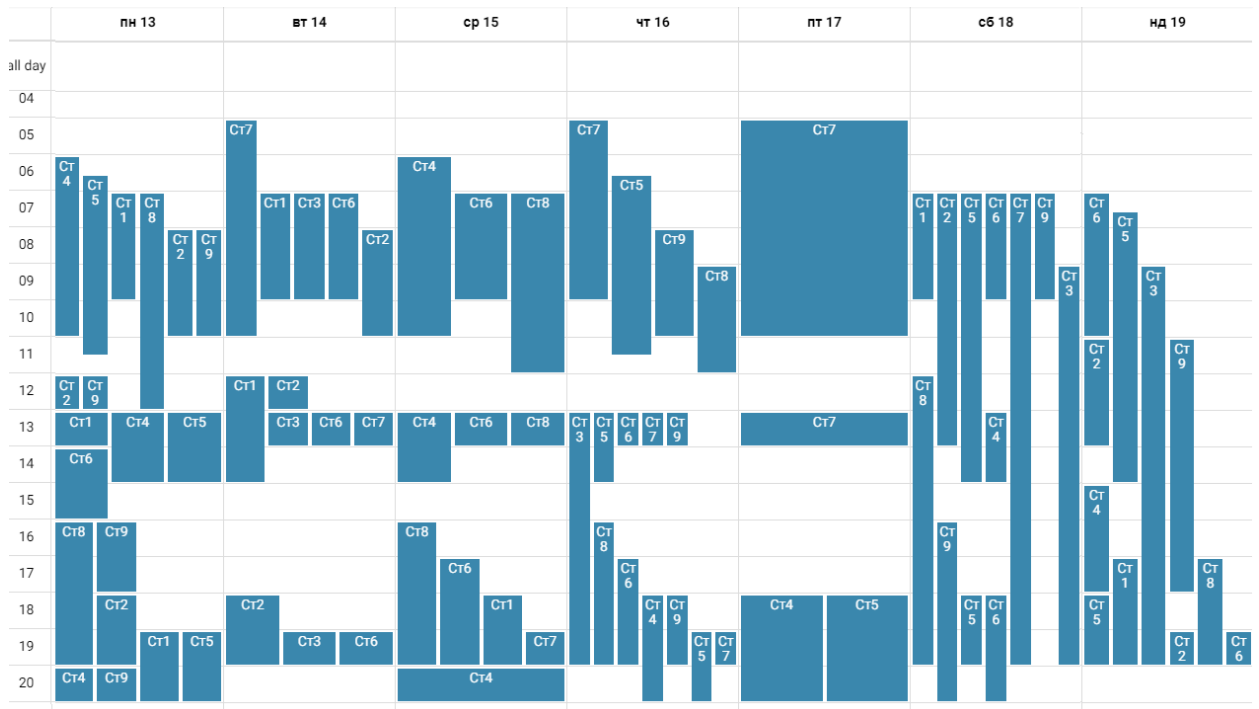


Рисунок 4.2 – Вихідні дані комбінованого методу: проміжки вільного часу студентів

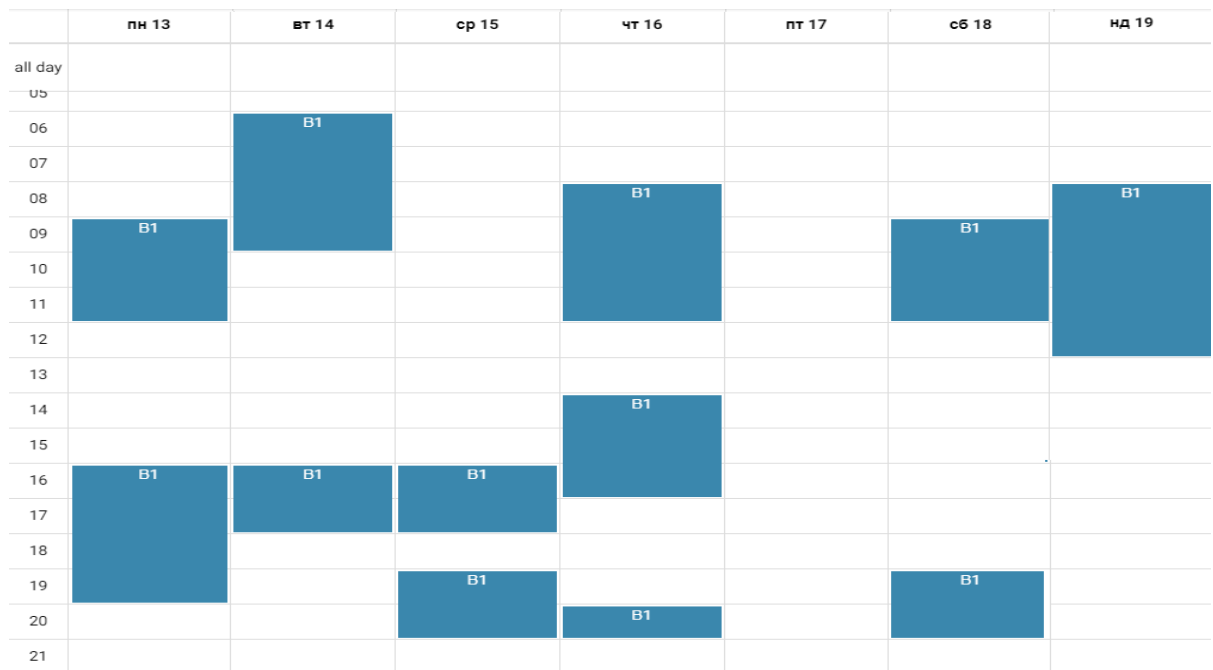


Рисунок 4.3 – Вихідні дані комбінованого методу: проміжки вільного часу викладача

## 4.3.2 Етап комбінованого методу «1. Формування обмежень»

Обмеження, що пов'язані з особливостями діяльності фірми подано у таблиці 4.5.

Таблиця 4.5 – Обмеження фірми дистанційного навчання

Х – множина змінних	D – область значень	С – множина обмежень
навантаження викладача «Викладач 1»	$0 \leq \text{кількість годин на тиждень} \leq 40 \text{ годин}$	$9 \text{ занять} * 45 \text{ хв.} = \{6 \text{ годин } 45 \text{ хв.}\}$
тривалість уроку	45	{45}
тривалість перерви викладача	15	{15}
період часу, на який створюється розклад	період часу, що належить відрізку, де початок – найбільш рання дата початку з курсів для яких складається розклад, кінець – найпізніша дата закінчення	тиждень
кількість занять курсу на період	$0 \leq \text{кількість занять курсу на період} \leq \text{загальна кількість занять}$	{2, 3, 4}
проміжки часу викладача	$0 \leq \text{проміжки часу викладача} \leq \text{тиждень}$ (величина періоду створення розкладу)	{таблиця 4.2}

Критерії значущості використовуються базові, тобто кількість занять курсу з вагою – 2, та кількість співпадінь вподобань – 1.

Формально визначені умови:

- оцінювання завершення ітерації: кількість занять розкладу створених в ітерації дорівнює бажаній, тобто, встановленій в сутності курсу;
- умови оцінювання успішного створення розкладу: кількість створених занять всіх курсів дорівнює бажаній, тобто запланованій.

#### 4.3.3 Етап комбінованого методу «2. Підготовка вхідних даних на основі вводу користувача»

Етап «Підготовка вхідних даних на основі вводу від користувачів» займається перетворенням останнього рівня ієрархічної структури.

По-перше, формується вигляд базової сутності проміжку, в тому вигляді, в якому він зберігається в системі.

Базова сутність проміжку складається з наступних полів: «Старт», «Кінець», «Ідентифікатор користувача», «Ідентифікатор проміжку».

Кінцевий варіант сутності проміжку: «Старт», «Кінець», «Кількість співпадінь вподобань».

Базовий та кінцевий варіант сутності проміжку за комбінованим методом для фірми дистанційних курсів наведено на рис 4.4.

Базова сутність проміжку	Кінцева сутність проміжку										
<table border="1" data-bbox="308 1335 762 1709"> <thead> <tr> <th data-bbox="308 1335 762 1395">Проміжок</th> </tr> </thead> <tbody> <tr> <td data-bbox="308 1395 762 1496">Ідентифікатор проміжку</td> </tr> <tr> <td data-bbox="308 1496 762 1597">Ідентифікатор користувача</td> </tr> <tr> <td data-bbox="308 1597 762 1653"><b>Старт</b></td> </tr> <tr> <td data-bbox="308 1653 762 1709"><b>Кінець</b></td> </tr> </tbody> </table>	Проміжок	Ідентифікатор проміжку	Ідентифікатор користувача	<b>Старт</b>	<b>Кінець</b>	<table border="1" data-bbox="930 1352 1390 1630"> <thead> <tr> <th data-bbox="930 1352 1390 1413">Проміжок</th> </tr> </thead> <tbody> <tr> <td data-bbox="930 1413 1390 1469"><b>Старт</b></td> </tr> <tr> <td data-bbox="930 1469 1390 1525"><b>Кінець</b></td> </tr> <tr> <td data-bbox="930 1525 1390 1581">Кількість</td> </tr> <tr> <td data-bbox="930 1581 1390 1630"><b>співпадінь вподобань</b></td> </tr> </tbody> </table>	Проміжок	<b>Старт</b>	<b>Кінець</b>	Кількість	<b>співпадінь вподобань</b>
Проміжок											
Ідентифікатор проміжку											
Ідентифікатор користувача											
<b>Старт</b>											
<b>Кінець</b>											
Проміжок											
<b>Старт</b>											
<b>Кінець</b>											
Кількість											
<b>співпадінь вподобань</b>											

Рисунок 4.4 – Базовий та кінцевий варіант сутності проміжку комбінованого методу для фірми дистанційних курсів

За алгоритм перетворень проміжків береться базовий, з наступними етапами: «1. Проміжки вільного часу», «2.2. Фільтрація на основі проміжків викладачів», «3.1. Злиття проміжків з вказанням кількості співпадінь», «3.2. Фільтрація співпадінь з вже створеними уроками». Пункт «2.1. Створення додаткових проміжків ШНМ» – в експерименті не розглядається.

«2.2. Фільтрація на основі проміжків викладачів» - відбувається наступним чином: ті проміжки, що не мають перетину з проміжками часу викладача – видаляються, ті що мають частковий перетин – утискаються, повний – переходять далі. Відфільтровані проміжки студентів у календарному представленні наведено на рис 4.4.

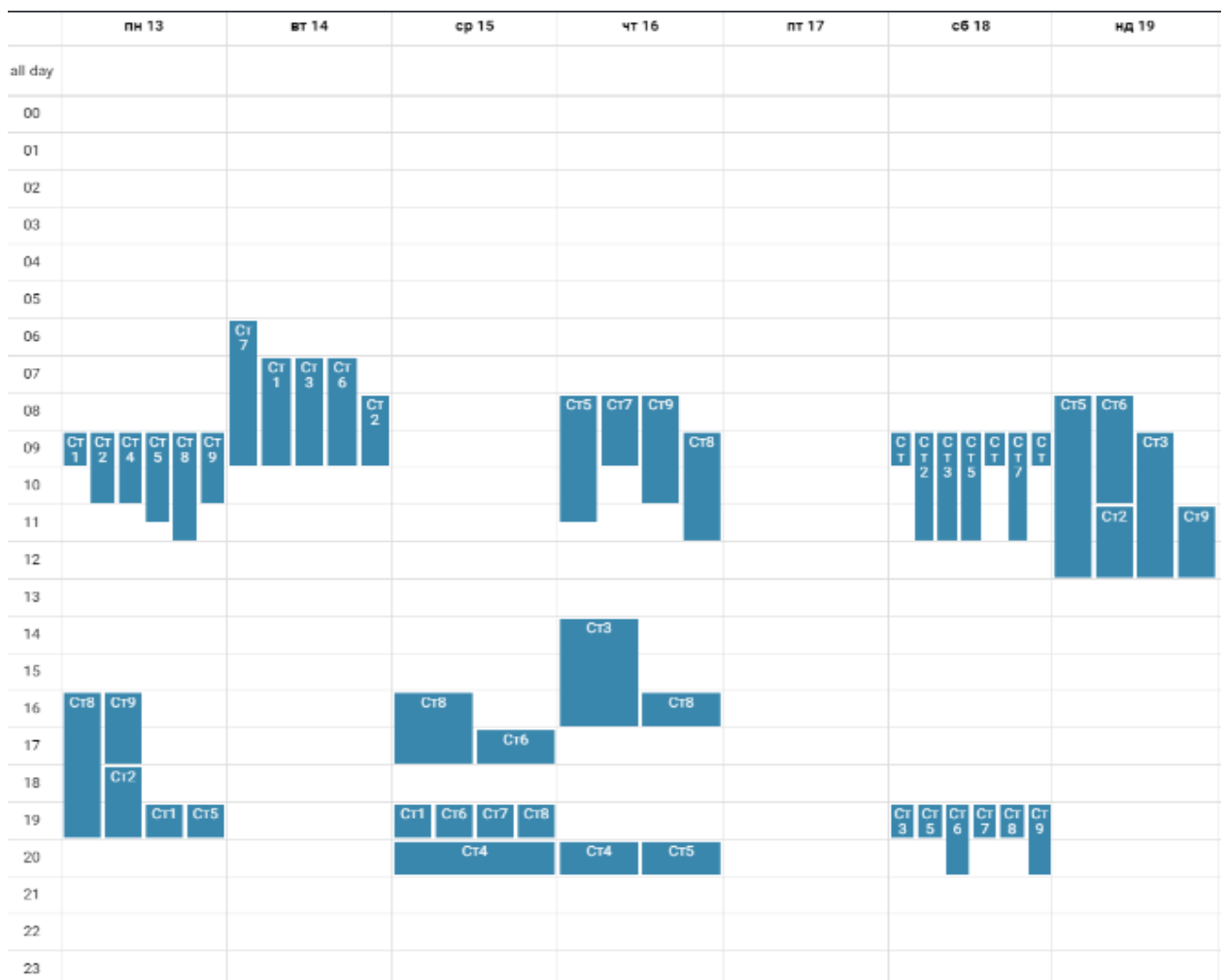


Рисунок 4.5 – Результати під-етапу перетворення проміжків вільного часу «Фільтрація на основі проміжків викладачів» комбінованого методу

Данні проміжків студентів після під-етапу перетворення проміжків вільного часу «Фільтрація на основі проміжків викладачів» комбінованого методу подано у таблиці 4.6.

Таблиця 4.6 – Данні проміжків студентів після під-етапу «Фільтрація на основі проміжків викладачів» комбінованого методу

№	Проміжки						
	Пн	Вт	Ср	Чт	Пт	Сб	Нед
1	09:00-09:45 19:00-20:00	07:00-09:45	19:00-20:00	-	-	09:00-10:00	-
2	09:00-10:45 18:00-20:00	08:00-10:00	-	-	-	09:00-12:00	11:00-13:00
3	-	07:00-09:45	-	14:00-17:00	-	09:00-12:00 19:00-20:00	09:00-13:00
4	09:00-11:00	-	20:00-21:00	20:00-21:00	-	-	-
5	09:00-11:30 19:00-20:00	-	-	08:00-11:30 20:00-21:00	-	09:00-12:00 19:00-20:00	08:00-13:00
6	-	07:00-09:45	17:00-18:00 19:00-20:00	-	-	09:00-10:00 19:00-21:00	08:00-11:00
7	-	06:00-10:00	19:00-20:00	08:00-09:45	-	09:00-12:00 19:00-20:00	-
8	09:00-12:00 16:00-20:00	-	16:00-18:00 19:00-20:00	09:00-11:45 16:00-17:00	-	19:00-20:00	-
9	09:00-10:45 16:00-18:00	-	-	08:00-10:45	-	09:00-10:00 19:00-21:00	11:00-13:00

З таблиці видно, що більшість проміжків була стиснута чи видалена, такий вид фільтрування засновано на тому, що викладач виставляє так багато часу як має змогу і в періоди часу не вказані – не може вести заняття.

Під-етап «3.1. Злиття проміжків з вказанням кількості співпадінь» використовує базовий алгоритм злиття проміжків. Тобто, кожний проміжок, що має перетин з іншим отримує плюс один до кількості співпадінь вподобань – скільки перетинів стільки й кількості співпадінь. Найбільш стиснутий перетин – буде мати найбільш великий пріоритет. Одинакові проміжки зливаються, а їх кількості підсумовуються.

Злиття проміжків відбувається з урахуванням отриманою ієрархічної структури. Ієрархічна структура розкладу ІТ-проекту дистанційного навчання наведено на рис. 4.6.

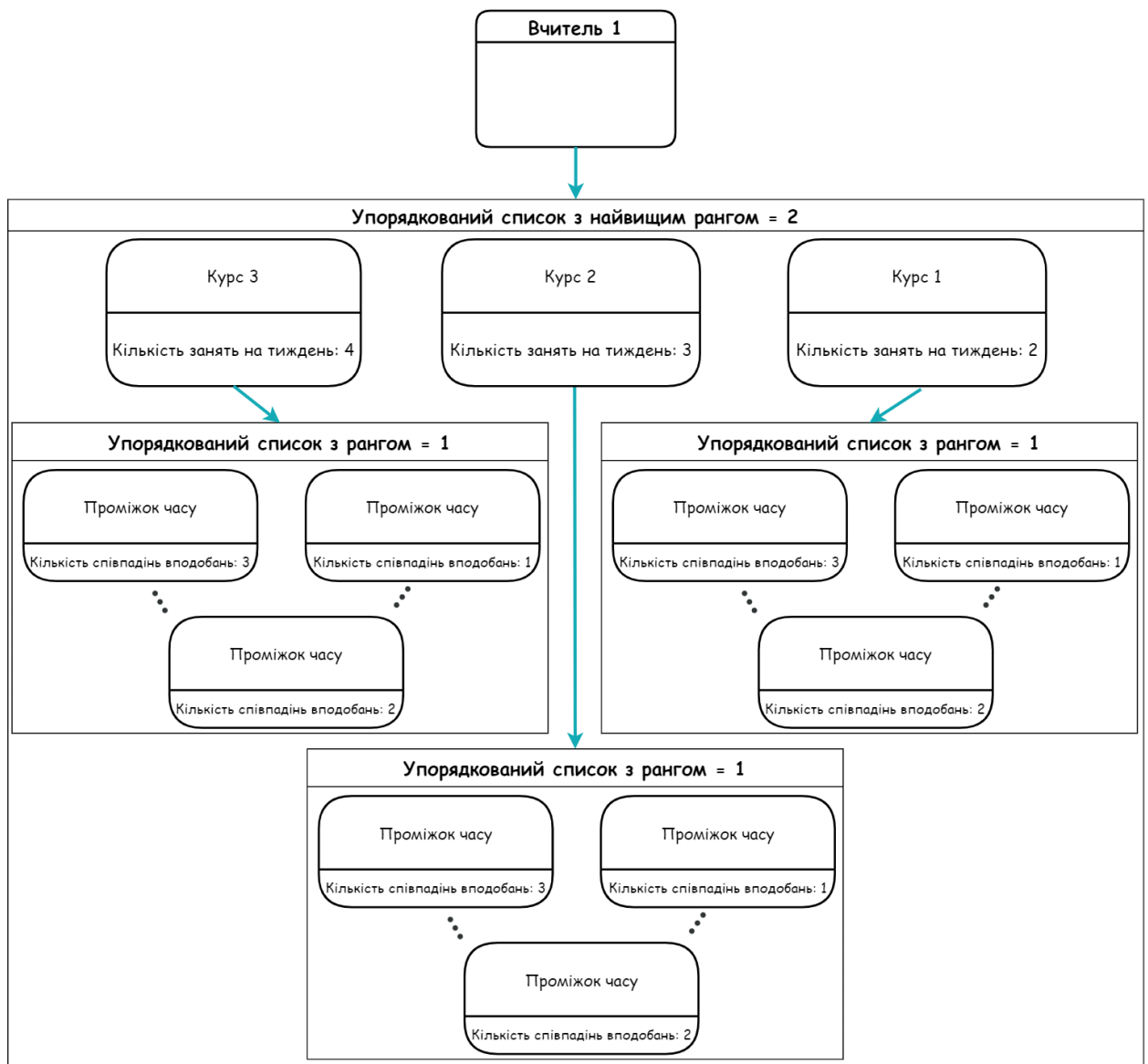


Рисунок 4.6 – Ієрархічна структура розкладу за комбінованим методом

Відсортовані та злиті проміжки для «Курс 3» наведено у таблиці 4.7, «Курс 2» - таблиці 4.8, «Курс 1» - таблиці 4.9.

Таблиця 4.7 – Кінцеві проміжки за комбінованим методом для «Курс 3»

День неділі	Проміжок	Кількість співпадінь вподобань
четвер	08:00-09:45	3
субота	19:00-20:00	3
понеділок	09:00-12:00	2
понеділок	16:00-20:00	2
середа	19:00-20:00	2
четвер	09:00-11:45	2
субота	09:00-12:00	2
понеділок	09:00-10:45	1
понеділок	16:00-18:00	1
вівторок	06:00-10:00	1
середа	16:00-18:00	1
четвер	08:00-10:45	1
четвер	16:00-17:00	1
субота	09:00-10:00	1
субота	19:00-21:00	1
неділя	11:00-13:00	1

Таблиця 4.8 – Кінцеві проміжки за комбінованим методом для «Курс 2»

День неділі	Проміжок	Кількість співпадінь вподобань
понеділок	09:00-11:00	2
середа	20:00-21:00	2
четвер	20:00-21:00	2
субота	09:00-12:00	2
субота	19:00-20:00	2
неділя	08:00-13:00	2
понеділок	09:00-11:30	1
понеділок	19:00-20:00	1
вівторок	07:00-9:45	1
середа	17:00-18:00	1
середа	19:00-20:00	1
четвер	08:00-11:30	1
субота	09:00-10:00	1
субота	19:00-21:00	1
неділя	08:00-11:00	1

Таблиця 4.9 – Кінцеві проміжки за комбінованим методом для «Курс 1»

День неділі	Проміжок	Кількість співпадінь вподобань
вівторок	07:00-09:45	3
субота	09:00-10:00	3
понеділок	09:00-09:45	2
понеділок	19:00-20:00	2
вівторок	08:00-10:00	2
субота	09:00-12:00	2
неділя	11:00-13:00	2
понеділок	09:00-10:45	1
понеділок	18:00-20:00	1
серeda	19:00-20:00	1
четвер	14:00-17:00	1
субота	19:00-20:00	1
неділя	09:00-13:00	1

#### 4.3.4 Етап комбінованого методу «3. Формування розкладу», під-етап «3.1. Пріоритизація занять»

Результати роботи жадібного алгоритму – гнучкий розклад занять з використанням базового алгоритму «3.2. Фільтрація співпадінь з вже створеними уроками» та умовами завершення ітерації, умовами оцінювання успішного створення розкладу представлено у таблиці 4.10.

Таблиця 4.10 – Результат комбінованого методу створення розкладу

Назва курсу	Розклад
Курс 3	Заняття 1 – понеділок 09:00-09:45 Заняття 2 – серeda 19:00-19:45 Заняття 3 – четвер 8:00-8:45 Заняття 4 – субота 19:00-19:45
Курс 2	Заняття 1 – понеділок 10:00-10:45 Заняття 2 – серeda 20:00-20:45 Заняття 3 – четвер 20:00-20:45
Курс 1	Заняття 1 – вівторок 07:00-07:45 Заняття 2 – субота 09:00-09:45

Отже, отриманий розклад представляє собою – розклад на основі компромісів між проміжками вільного часу студентів. При цьому, проміжки вільного часу викладача виступають у якості обмеження розкладу.

Заняття створюються в порядку найбільшого пріоритету курсу, тобто за визначеними критеріями ефективності та їхніми вагами, ієрархічною структурою – значенням кількості занять на тиждень.

Створенні заняття на попередній ітерації враховуються – перетини з ними видаляють, при цьому зберігається задана в системі перерва, що складає 15 хвилин.

Календарне представлення створеного розкладу наведено на рис 4.7.

	пн 13	вт 14	ср 15	чт 16	пт 17	сб 18
all day						
06						
07		К-1				
08				К-3		
09	К-3					К-1
10	К-2					
11						
12						
13						
14						
15						
16						
17						
18						
19			К-3			К-3
20			К-2	К-2		
21						
22						
23						

Рисунок 4.7 – Розклад за комбінованим методом

4.3.5 Етап комбінованого методу «3. Формування розкладу», під-етап «3.2. Обробка конфліктів»

Порівняльний аналіз задоволеності ГРЗ, порівняно з ФРЗ для «Курс 3» наведено у таблиці 4.11, «Курс 2» – таблиці 4.12, «Курс 1» – таблиці 4.13.

Таблиця 4.11 – Порівняльний аналіз задоволеності розкладом «Курс 3»

Користувач	Заняття 1		Заняття 2		Заняття 3		Заняття 4		Результат	
	ГРЗ	ФРЗ	ГРЗ	ФРЗ	ГРЗ	ФРЗ	ГРЗ	ФРЗ	ГРЗ	ФРЗ
Викладач 1	1	1	1	0	1	0	1	0	100%	25%
Студент 9	1	0	0	0	1	1	1	0	75%	25%
Студент 8	1	1	1	0	1	1	1	0	100%	50%
Студент 7	0	0	1	0	1	1	1	0	75%	25%

Таблиця 4.12 – Порівняльний аналіз задоволеності розкладом «Курс 2»

Користувач	Заняття 1		Заняття 2		Заняття 3		Результат	
	ГРЗ	ФРЗ	ГРЗ	ФРЗ	ГРЗ	ФРЗ	ГРЗ	ФРЗ
Викладач 1	1	1	1	1	1	0	100%	66,6%
Студент 6	0	0	0	1	0	0	0%	33,3%
Студент 5	1	0	0	0	1	0	33,3%	0%
Студент 4	1	0	1	0	1	0	100%	0%

Таблиця 4.13 – Порівняльний аналіз задоволеності розкладом «Курс 1»

Користувач	Заняття 1		Заняття 2		Результат	
	ГРЗ	ФРЗ	ГРЗ	ФРЗ	ГРЗ	ФРЗ
Викладач 1	1	0	1	1	100%	50%
Студент 3	1	1	1	1	100%	100%
Студент 2	1	1	1	1	100%	100%
Студент 1	1	1	1	0	100%	50%

Під-етап комбінованого методу створення розкладу «3.2 Обробка конфліктів», включає в себе аналіз результатів. За формально вираженою умовою «Чи є студент з 0% задоволеності розкладом?» - якщо так – «Чи є проміжки, які б зрівняли задоволеність студентів?».

За результатами під-етапу комбінованого методу, заняття 3 буде перенесено на неділю 8:00-8:45. Порівняльний аналіз задоволеності оновленим розкладом для «Курс 2» наведено у таблиці 4.14.

Таблиця 4.14 – Порівняльний аналіз задоволеності оновленим розкладом для «Курс 2»

Користувач	Заняття 1		Заняття 2		Заняття 3		Результат	
	ГРЗ	ФРЗ	ГРЗ	ФРЗ	ГРЗ	ФРЗ	ГРЗ	ФРЗ
Викладач 1	1	1	1	1	1	0	100%	66,6%
Студент 6	0	0	0	1	1	0	33,3%	33,3%
Студент 5	1	0	0	0	1	0	33,3%	0%
Студент 4	1	0	1	0	0	0	33,3%	0%

Оновлений розклад за комбінованим методом у календарному вигляді наведено на рис. 4.8.

	нд 12	пн 13	вт 14	ср 15	чт 16	пт 17	сб 18	нд 19
06								
07			К-1					
08					К-3			К-2
09		К-3					К-1	
10		К-2						
11								
12								
13								
14								
15								
16								
17								
18								
19				К-3			К-3	
20				К-2				

Рисунок 4.8 – Оновлений розклад занять за комбінованим методом

Данні занять оновленого розкладу за комбінованим методом подано у таблиці. 4.15.

Таблиця 4.15 – Данні занять оновленого розкладу за комбінованим методом

Назва курсу	Розклад
Курс 3	Заняття 1 – понеділок 09:00-09:45 Заняття 2 – середа 19:00-19:45 Заняття 3 – четвер 8:00-8:45 Заняття 4 – субота 19:00-19:45
Курс 2	Заняття 1 – понеділок 10:00-10:45 Заняття 2 – середа 20:00-20:45 Заняття 3 – неділя 08:00-08:45
Курс 1	Заняття 1 – вівторок 07:00-07:45 Заняття 2 – субота 09:00-09:45

Підводячи підсумок, зведені результати порівняння задоволеності студентів розкладом наведено у таблиці 4.16.

Таблиця 4.16 – Зведені результати порівняння задоволеності студентів розкладом

Показник	ГРЗ (оновлений)	ФРЗ
Найнижче значення	33,3%	0%
Середнє значення «Курс 1»	100%	83,3%
Середнє значення «Курс 2»	33,3%	11,1%
Середнє значення «Курс 3»	83,3%	33,3%
Середнє значення за всіма курсами	72,2%	42,6%
Середнє значення за всіма курсами вчителя	100%	47,2%

Отже, задоволеність студентів гнучким розкладом у порівнянні з фіксованим у 1.69 разів вище. Для вчителя – це співвідношення становить 2,1 рази.

#### 4.4 Висновки з четвертого розділу кваліфікаційної роботи

В процесі апробації комбінованого методу та моделі підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання було зроблено наступне.

По-перше, результатом аналізу предметної галузі є схема організаційної структури об'єкту автоматизації, для кожної ланки структури вказано її обов'язки.

Також, перелічено особливості діяльності фірми дистанційних курсів, профіль типового клієнта, характеристика навантаження.

По-друге, розглянуто бізнес-процес створення розкладу, що реалізовано у фірми на поточний час, визначено мету використання комбінованого методу.

По-третє, проведено порівняння ГРЗ та ФРЗ, задля цього наведено приклад складання розкладу з вихідними даними, що відповідають особливостям роботи фірми дистанційних курсів. В ході вирішення задачі створення розкладу для фірми дистанційних курсів задокументовано усі етапи складання розкладу комбінованим методом.

За результатами апробації отримано такі дані: задоволеність студентів гнучким розкладом у порівнянні з фіксованим розкладом занять у 1.69 разів вище. Для вчителя – це співвідношення становить 2.1 рази.

Отже, в розділі було проведено апробацію комбінованого методу підвищення гнучкості розкладу ІТ проєкту дистанційного навчання.

За результатами розділу було отримано аналіз предметної галузі, сферу застосування методу, порівняно ФРЗ та ГРЗ за рівнем задоволеності розкладом у процентному вираженні.

## ВИСНОВКИ

За завданням магістерської кваліфікаційної роботи було проведено дослідження моделей і методів підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання, створено комбінований метод, що використовує сильні сторони існуючих, мінімізуючи при цьому їх недоліки.

У рамках магістерської кваліфікаційної роботи було доведено актуальність дослідження, проаналізовані існуючі моделі і методи підвищення гнучкості розкладу, існуючі рішення, обґрунтована мета розробки комбінованого методу та моделі, яка розглядається в цій роботі та сформовані вимоги, проблеми та критерії ефективності, поставлена задача дослідження. Робота оформлена згідно з стандартами [29, 30].

На основі цих досліджень був розроблений комбінований метод підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання, а саме – описано етапи створення розкладу, описано особливості використання всіх його складових частин: моделі ШНМ, алгоритму на основі обмежень, генетичного алгоритму та жадібного алгоритму. Розгляд складових частин алгоритму частин проводився з прив'язкою к етапам методу.

Проведено дослідження отриманих наукових результатів, а саме дослідження особливостей реалізації комбінованого методу і моделі, наведено сценарії використання комбінованого методу, описано взаємодію користувачів з ним, описано його реалізацію на платформи-незалежному рівні.

В якості апробації комбінованого методу задокументовано створення розкладу для фірми дистанційних курсів, на тестових даних.

Отже, усі завдання кваліфікаційної роботи виконані – дослідження моделей і методів підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання проведено, комбінований метод створено.

Результати кваліфікаційної роботи було опубліковано в роботі [2].

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Методичні вказівки щодо розробки та оформлення магістерської атестаційної роботи за спеціальністю 122 Комп'ютерні науки (освітня програма «Управління проектами в галузі інформаційних технологій» освітньо-кваліфікаційного рівня «магістр» / Упоряд.: Петров К.Е., Левикін В.М., Чалий С.Ф., Євланов М.В., Саєнко В.І., Міхнов Д.К., Міхнова А.В., Чала О.В. – Харків: ХНУРЕ, 2019. – 28 с.
2. Кудрявцева М.О. Дослідження моделей і методів підвищення гнучкості розкладу ІТ-проєкту дистанційного навчання. //27-й Міжнародний молодіжний форум «Радіоелектроніка та молодь ХХІ столітті». Зб. матеріалів форуму. Т.6. Конференція «Інформаційні інтелектуальні системи» – Харків: ХНУРЕ. 2023.
3. SWOT analysis: A tool for making better business decisions. / ed. by United States. Dept. of Agriculture. Risk Management Agency. [Washington, D.C.] : U.S. Dept. of Agriculture, Risk Management Agency, 2008.
4. Вивчати англійську мову онлайн на курсах Green Forest із будь-якого міста. GreenForest. URL: <https://greenforest.com.ua/courses/anhliiska-online> (дата звернення: 21.02.2023).
5. Create & share a group calendar - Google Workspace Admin Help. Google Help. URL: <https://support.google.com/a/answer/1626902?hl=en> (дата звернення: 21.02.2023).
6. Introduction to the Outlook Calendar - Microsoft Support. Microsoft Support. URL: <https://support.microsoft.com/en-us/office/introduction-to-the-outlook-calendar-d94c5203-77c7-48ec-90a5-2e2bc10bd6f8> (дата звернення: 21.02.2023).
7. PDF documentation - aSc Timetables. aSc TimeTables - Online Help. URL: <https://help.asctimetables.com/text.php?id=592&lang=en> (date of access: 21.02.2023).

8. Lister A. M., Eager R. D. Resource Allocation and Scheduling. London : Macmillan Education UK, 1993.
9. Cormier D., O'Grady P., Sanii E. A constraint-based genetic algorithm for concurrent engineering. International Journal of Production Research. 1998. Vol. 36, no. 6. P. 1679–1697.
10. Schmitt L. M. Theory of genetic algorithms. Theoretical Computer Science. 2001. Vol. 259, no. 1-2. P. 1–61.
11. Shanmuganathan S. Artificial Neural Network Modelling: An Introduction. Artificial Neural Network Modelling. Cham, 2016. P. 1–14.
12. Saaty T. L. Group Decision Making and the AHP. The Analytic Hierarchy Process. Berlin, Heidelberg, 1989. P. 59–67.
13. A Guide to the Project Management Body of Knowledge (PMBOK® Guide). Newtown Square, USA : Project Management Institute, 2021.
14. Buduma N. Fundamentals of Deep Learning: Designing Next-Generation Machine Intelligence Algorithms. O'Reilly Media, 2017. 298 p.
15. Di W., Bhardwaj A., Wei J. Deep Learning Essentials: Your hands-on guide to the fundamentals of deep learning and neural network modeling. Packt Publishing, 2018. 284 p.
16. Rungta K. TensorFlow in 1 Day: Make your own Neural Network. Independently Published, 2018. 364 p.
17. Trask A. W. Grokking Deep Learning. Manning Publications Co. LLC, 2019.
18. Yokoo M. Constraint Satisfaction Problem. Distributed Constraint Satisfaction. Berlin, Heidelberg, 2001. P. 1–45.
19. Bodirsky M., Pinsker M. Schaefer's Theorem for Graphs. Journal of the ACM. 2015. Vol. 62, no. 3. P. 1–52.
20. Kramer O. Genetic Algorithm Essentials. Cham : Springer International Publishing, 2017.
21. Mohammed Q. Mohammed, Saif Q. Muhamed, M. Ievlanov, Z. Gazetdinova. Improvement of the method of scenario analysis of functional

requirements to an information systems // Eastern-European Journal of Enterprise Technologies. – 2019. – № 3/2 (99). – pp. 25-35.

22. Patton J., Economy P. User Story Mapping: Discover the Whole Story, Build the Right Product. O'Reilly Media, Incorporated, 2014.

23. Larman C. Applying UML and Patterns: An Introduction to Object-oriented Analysis and Design and Iterative Development. Dorling Kindersley Pvt Ltd, 2008. 736 p.

24. Martin R. C. Agile Software Development, Principles, Patterns, and Practices. 2nd ed. Prentice Hall, 2002. 529 p.

25. Fowler M. Patterns of Enterprise Application Architecture. Addison-Wesley Professional, 2002. 560 p.

26. Господарський кодекс України. Офіційний вебпортал парламенту України. URL: <https://zakon.rada.gov.ua/laws/show/436-15#Text> (дата звернення: 14.03.2023).

27. Про захист інформації в інформаційно-комунікаційних системах. Офіційний вебпортал парламенту України. URL: <https://zakon.rada.gov.ua/laws/80/94-BP#Text> (дата звернення: 14.03.2023).

28. Про захист персональних даних. Офіційний вебпортал парламенту України. URL: <https://zakon.rada.gov.ua/laws/2297-17> (дата звернення: 14.03.2023).

29. ДСТУ 3008:2015. Інформація та документація. Звіти у сфері науки і техніки. Структура і правила оформлювання. – Чинний від 22.06.2015. – Київ: ДП «УкрНДНЦ», 2016. – 31 с.

30. ДСТУ 8302:2015. Інформація та документація. Бібліографічні посилання. Загальні положення та правила складання. – Чинний від 04.03.2016. – Київ: ДП «УкрНДНЦ», 2016. – 20 с.