Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)
Кафедра Інформатики
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

ДОСЛІДЖЕННЯ МЕТОДУ ГЕЙЛА-ШЕПЛІ ДЛЯ РОЗПОДІЛУ ДАНИХ У ЗАДАЧАХ ПРИЙНЯТТЯ РІШЕНЬ
(тема)

Виконав:
студент 2 курсу, групи ІНФм-22-2
Тарасенко Д.О.
(прізвище, ініціали)

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

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

Керівник доц. Творошенко І.С.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ Кобилін О.А.
(підпис) (прізвище, ініціали)

2024 р.

Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)

Кафедра Інформатики
(повна назва)

Рівень вищої освіти другий (магістерський)

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

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«____»_____ 2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Тарасенку Денису Олександровичу
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження методу Гейла-Шеплі для розподілу даних у задачах прийняття рішень

затверджена наказом по університету від 3 листопада 2023 року № 1280Ст

2. Термін подання студентом роботи до екзаменаційної комісії 23 грудня 2023 р.

3. Вихідні дані до роботи науково-методична література, матеріали конференцій, теоретичні відомості методів для розподілу даних, теоретичні відомості про алгоритм Гейла-Шеплі, використовувані програмні засоби: мова програмування TypeScript, бібліотека React.js, фреймворк NextJS, Material-UI, СУБД PostgreSQL.

4. Перелік питань, що потрібно опрацювати в роботі
1. Огляд методів детектування розподілу даних у задачах прийняття рішень.
2. Аналіз методу Гейла-Шеплі.
3. Розробка алгоритму Гейла-Шеплі.
4. Розробка та реалізація програмного застосунку для розподілу волонтерів між волонтерськими проєктами, в основі якого застосовано метод Гейла-Шеплі.
5. Тестування розробленого застосунку.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) актуальність проблеми, постановка задачі, схема роботи алгоритму, етапи реалізації поставленої задачі.

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

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

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	03.11.2023	
2	Аналіз завдання, підбір літератури	03.11.23-05.11.23	
3	Аналіз літератури з досліджуваної проблеми	05.11.23-07.11.23	
4	Аналіз інструментальних засобів	07.11.23-09.11.23	
5	Розробка методу та концепції застосунку	09.11.23-11.11.23	
6	Програмна реалізація	11.11.23-16.11.23	
7	Оформлення пояснювальної записки	16.12.23-26.11.23	
8	Перевірка на плагіат	06.12.2023	
9	Рецензування	10.12.2023	
10	Підготовка презентації та доповіді	17.12.2023	
11	Занесення роботи в електронний архів	02.01.2024	
12	Попередній захист кваліфікаційної роботи	02.01.2024	

Дата видачі завдання 3 листопада 2023 р.

Студент _____
(підпис)

Керівник роботи _____ доц. Творошенко І.С.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 74 с., 2 табл., 15 рис., 1 дод., 41 джерело.

СТАБІЛЬНІ ВІДПОВІДНОСТІ, АЛГОРИТМ ГЕЙЛА-ШЕПЛІ, ЗАДАЧА МАР'ЯЖУ, ПРИЙНЯТТЯ РІШЕНЬ, РОЗПОДІЛ ДАНИХ, ТЕОРІЯ ВІДПОВІДНОСТІ.

Об'єктом дослідження є процес розроблення застосунку для розподілу волонтерів між волонтерськими проєктами.

Метою дослідження є розробка застосунку для розподілу волонтерів між волонтерськими проєктами на основі методу Гейла-Шеплі, що призначений для створення стабільних відповідностей.

Проведено аналіз існуючих методів для розподілу даних у задачах прийняття рішень. Проведено дослідження алгоритму Гейла-Шеплі, його модифікацій.

У результаті дослідження здійснена програмна реалізація застосунку для розподілу волонтерів між волонтерськими проєктами з використанням алгоритму Гейла-Шеплі.

STABLE MATCHING, GALE-SHAPLEY ALGORITHM, MARRIAGE PROBLEM, DECISION MAKING, DATA DISTRIBUTION, MATCHING THEORY.

The object of research is the process of developing an application for the distribution of volunteers between volunteer projects.

The research aims to develop an application for assigning volunteers to volunteer projects based on the Gale-Shapley method, which is designed to create stable matches.

An analysis of existing methods for data distribution in decision-making tasks was carried out. A study of the Gale-Shapley algorithm and its modifications was conducted.

As a result of the research, a software implementation of the application for the distribution of volunteers between volunteer projects using the Gale-Shepley algorithm was carried out.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів 5
Вступ 9
1 Аналіз існуючих застосунків для розподілу даних у задачах прийняття рішень 10
1.1 Аналіз сучасних застосунків для розподілу даних у задачах прийняття рішень 10
1.2 Класифікація та аналіз існуючих методів для розподілу даних у задачах прийняття рішень 14
1.2.1 Методи випадкової вибірки 14
1.2.2 Методи повторної вибірки 17
1.2.3 Кластерні методи 18
1.2.4 Методи поділу даних 21
1.2.5 Методи зменшення розмірності 22
1.2.6 Алгоритми зіставлення 24
1.2.7 Алгоритми дерева рішень 25
1.3 Особливості задач прийняття рішень 28
1.4 Аналіз літературних джерел щодо апробації результатів стосовно розподілу даних у задачах прийняття рішень 30
1.5 Постановка задачі дослідження 33
2 Особливості методу Гейла-Шеплі для розподілу даних у задачах прийняття рішень 35
2.1 Особливості методу Гейла-Шеплі 35
2.1.1 Переваги та недоліки алгоритму Гейла-Шеплі 36
2.1.2 Модифікації алгоритму Гейла-Шеплі 38
2.2 Механізм розподілу даних у задачах прийняття рішень 39
2.2.1 Збір даних 39
2.2.2 Попередня обробка даних 41
2.2.3 Аналіз даних 42

2.2.4 Припущення щодо розподілу та розробка моделі розподілу даних 45
2.2.5 Прийняття рішень 47
2.2.6 Підтримка прийняття рішень і моніторинг 49
2.3 Методика розподілу даних у задачах прийняття рішень на основі методу Гейла-Шеплі 50
3 Дослідження методу Гейла-Шеплі для розподілу даних у задачах прийняття рішень 53
3.1 Вибір інструментальних засобів для реалізації методу Гейла-Шеплі стосовно розподілу даних у задачах прийняття рішень 53
3.2 Етапи програмної реалізації методу Гейла-Шеплі для розподілу даних у задачах прийняття рішень 54
3.2.1 Опис концепції застосунку 54
3.2.2 Програмна реалізація 58
3.3 Тестування розробленого застосунку та аналіз результатів 63
3.4 Перспективи подальшої роботи 65
Висновки 66
Перелік джерел посилання 67
Додаток А Скріншоти сторінок застосунку 72

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

БД – база даних

БПЛА – безпілотні летальні апарати

API – Application Programming Interface (прикладний програмний інтерфейс)

BI – Business Intelligence (бізнес-аналітика)

CART – Classification And Regression Tree (дерева класифікації та регресії)

DBSCAN – Density-Based Spatial Clustering of Applications with Noise (просторова кластеризація застосунків із шумом на основі щільності)

EDA – Exploratory Data Analysis (розвідувальний аналіз)

FCM – Fuzzy *C*-Means (метод нечіткої кластеризації *c*-середніх)

GMM – Gaussian Mixture Models (моделі сумішей Гауса)

HAP – High Altitude Platforms (висотні платформи)

ID3 – Iterative Dichotomiser 3 (ітеративний дихотомізер 3)

LDA – Linear Discriminant Analysis (лінійний дискримінантний аналіз)

LightGBM – Light Gradient Boosting Machine

LP – Linear Prediction (лінійне передбачення)

LP-ALG – LP-based online algorithm (онлайн-алгоритм на основі LP)

npm – node package manager (менеджер пакетів вузлів)

OSM-KIID – Online Stable Matching under Known Identical Independent Distributions (стабільне зіставлення онлайн за відомих ідентичних незалежних розподілів)

PCA – Principal Component Analysis (аналіз основних компонентів)

PPS – Probability Proportional to Size (ймовірність, пропорційна розміру)

ROC – Receiver Operating Characteristic

ROC AUC – Area Under the ROC Curve

SAS – Statistical Analysis System

SPSS – Statistical Package for Social Science

SMOTE – Synthetic Minority Oversampling Technique (техніка надмірної вибірки синтетичної меншості)

t-SNE – t-distributed Stochastic Neighbor Embedding (t-розподілене стохастичне вбудовування сусідів)

XGBoost – eXtreme Gradient Boosting

ВСТУП

За останні десятиліття наука про прийняття рішень досягла значних успіхів. Проте, незважаючи на ці досягнення, проблеми прийняття рішень продовжують бути актуальними, а сам процес прийняття рішень досі є далеко не простий.

Щодня люди, організації та уряди стикаються з вибором, який може визначити їхні долі. Однією із найпоширеніших проблем є проблема створення стабільної відповідності між двома наборами елементів з різними перевагами для кожного з них. Сюди відносяться й розподіл вакантних робочих місць серед кандидатів, пошук кращої відповідності між орендарем та орендодавцем, розподіл медичних спеціалістів між закладами охорони здоров’я тощо. В кожному з цих прикладів є 2 набори сутностей зі своїми наборами пріоритетів з сутностей протилежного набору.

Рішення проблеми стабільних відповідностей у 1962 році запропонували Девід Гейл і Ллойд Шеплі. В своїй роботі «College Admissions and the Stability of Marriage» [1] автори досліджують застосування математичної концепції під назвою «проблема стабільного шлюбу» до контексту вступу до коледжу. Проблема стабільного шлюбу – це класична алгоритмічна задача, яка прагне знайти стабільну відповідність між двома наборами індивідів, наприклад чоловіками та жінками. Автори представили алгоритм Гейла-Шеплі, який є ефективним способом знайти стабільну відповідність між двома наборами агентів із перевагами, та довели, що для будь-якої рівної кількості учасників кожного типу завжди можна знайти відповідність, у якій усі пари стабільні [1].

Актуальність досліджень у цьому напрямку полягає в тому, що застосування алгоритму Гейла-Шеплі може допомогти в розв’язанні проблем адекватного розподілу обмежених ресурсів в різних сферах, таких як економіка, інформаційні технології, медицина та інші.

1 АНАЛІЗ ІСНУЮЧИХ ЗАСТОСУНКІВ ДЛЯ РОЗПОДІЛУ ДАНИХ У ЗАДАЧАХ ПРИЙНЯТТЯ РІШЕНЬ

1.1 Аналіз сучасних застосунків для розподілу даних у задачах прийняття рішень

В сучасному світі існує багато застосунків та інструментів, які можна використовувати для розподілу даних у задачах прийняття рішень [2]. Ці програми створені, щоб допомогти організаціям і окремим особам збирати, аналізувати та поширювати дані для підтримки прийняття обґрунтованих рішень. Нижче наведено короткий аналіз деяких з них.

Microsoft Excel – це широко використовувана програма для роботи з електронними таблицями, яку можна використовувати для аналізу та розподілу даних. Він підходить для малих і середніх наборів даних і може створювати діаграми та графіки для візуалізації даних. Microsoft Excel має такі сильні сторони, як: широкодоступність, зручність і універсальність для малих і середніх наборів даних. Пропонує основні можливості аналізу даних, побудови діаграм і обчислень. Проте має обмежену масштабованість для обробки великих або складних наборів даних. Не дуже підходить для співпраці в реальному часі.

Подібно до Excel, Google Таблиці – це хмарна програма для роботи з електронними таблицями, яка дозволяє спільно аналізувати дані та обмінюватися ними. Декілька користувачів можуть працювати над одним документом одночасно. З переваг можна виділити те, що це хмарний застосунок, зі спільним доступом і безкоштовний. Кілька користувачів можуть працювати над одним документом у режимі реального часу. Підходить для базового аналізу та обміну даними. З недоліків те, що може не вистачати деяких розширених функцій у спеціальних інструментах аналізу даних. Як і Excel, він може погано працювати з великими наборами даних

або складним аналізом. Розширені функції візуалізації даних обмежені порівняно зі спеціальними інструментами BI.

Tableau – це потужний інструмент візуалізації даних, який допомагає користувачам створювати інтерактивні інформаційні панелі, якими можна ділитися. Це особливо корисно для створення візуально привабливих звітів і візуалізацій. Інструмент має виняткові можливості візуалізації даних, інтерактивні інформаційні панелі та широкий спектр з'єднувачів даних. Підтримує поглиблений аналіз і оповідання даних. Проте може бути дорогим для окремих користувачів або малого бізнесу. Необхідно значно більше часу для навчання складним візуалізаціям.

Microsoft Power BI – це служба бізнес-аналітики, яка забезпечує інтерактивні візуалізації та можливості бізнес-аналітики з простим у використанні інтерфейсом. Він може підключатися до широкого кола джерел даних і підходить для обміну інформацією з іншими. З переваг – інтеграція з екосистемою Microsoft, потужне моделювання даних, інтерактивні звіти та інформаційні панелі. Добре підходить для обміну інформацією між організаціями. Але для оптимального використання потрібне середовище на основі Microsoft. Вартість ліцензування може бути непомірно високою для деяких користувачів. Розширені функції можуть потребувати досвіду.

Інструменти QlikView і Qlik Sense пропонують візуалізацію даних і рішення бізнес-аналітики, які дозволяють користувачам досліджувати дані, створювати інтерактивні інформаційні панелі та ділитися інформацією. Сильними сторонами є: зручна, асоціативна модель даних і гарні можливості дослідження даних. Підходить для створення інтерактивних панелей самообслуговування. Натомість витрати на ліцензування можуть бути високими, і QlikView має крутішу криву навчання порівняно з більш зручними інструментами.

RStudio – це інтегроване середовище розробки для мова програмування R, що використовується статистичних обчислень і графіки [3]. Ці інструменти популярні серед аналітиків даних і статистиків для

поглибленого аналізу та візуалізації даних. Середовище надає можливість робити статистичний аналіз, маніпулювання даними та має гарні можливості візуалізації. Пропонує широкий вибір пакетів для спеціалізованих завдань. Проте для використання можуть знадобитися навички програмування та є крива навчання, може бути не найкращим вибором для швидкого, спеціального аналізу. Візуалізація даних може бути не такою зручною для користувача, як у спеціальних інструментах BI.

Python – це універсальна мова програмування з такими бібліотеками, як pandas і matplotlib для аналізу та візуалізації даних [4-6]. Jupyter Notebooks надають інтерактивне середовище для роботи з кодом і даними Python. Ноутбуки Jupyter забезпечують інтерактивне середовище, яким можна ділитися. Але подібно до R, потрібні навички програмування, і візуалізація даних може потребувати додаткових бібліотек і кодування.

IBM Cognos Analytics – це інструмент бізнес-аналітики корпоративного рівня, який дозволяє користувачам створювати та поширювати звіти та інформаційні панелі для підтримки прийняття рішень. Пропонує широку інтеграцію даних і функції керування. Але може бути дорогим і складним у налаштуванні. Найкраще підходить для великих організацій.

Статистичний програмний пакет SAS пропонує набір аналітичних інструментів SAS Analytics для керування даними, розширеної аналітики та звітності. Він зазвичай використовується в таких галузях, як охорона здоров’я та фінанси. Пакет надає потужні можливості аналізу даних і звітності, особливо в регульованих галузях. Підтримує розширене статистичне моделювання та аналітику. Проте має високі витрати на ліцензування та крутіша крива навчання порівняно з альтернативами з відкритим кодом.

D3.js – це бібліотека JavaScript для створення інтерактивних і настроюваних візуалізацій даних в Інтернеті. Вона дуже гнучка і може використовуватися для створення спеціальних рішень для розподілу даних. Перевагами є можливість налаштування та гнучкість для створення

інтерактивних вебвізуалізацій даних. Ідеально підходить для розробників і спеціалістів із обробки даних. Проте потрібне знання JavaScript, і це не повний інструмент аналізу даних.

Apache Superset – це платформа дослідження та візуалізації даних з відкритим кодом, яка інтегрується з різними джерелами даних. Це дозволяє користувачам створювати інтерактивні інформаційні панелі та ділитися ними з іншими. З недоліків – може знадобитися технічний досвід для розширеного налаштування, можуть бути відсутні деякі розширені функції комерційних інструментів BI, а також зручність для користувача різна.

Looker – це платформа даних, яка надає інструменти для дослідження даних, візуалізації та співпраці. Її часто використовують для створення звітів і інформаційних панелей на основі даних і обміну ними. Добре інтегрується з різними джерелами даних. Але витрати на ліцензування можуть бути високими, і платформа може не запропонувати стільки налаштувань, скільки інші інструменти, такі як D3.js.

Metabase – це інструмент бізнес-аналітики та аналітики з відкритим кодом, який спрощує дослідження даних і звітування. Він зручний і підходить для невеликих команд і організацій. Проте він може не мати всіх розширених функцій комерційних інструментів BI та може бути погано масштабованим для великих підприємств.

Вибираючи програму для розподілу даних під час прийняття рішень, необхідно враховувати різні фактори, такі як: розмір і складність даних, технічні навички команди, вимоги до співпраці та бюджетні обмеження [2]. Крім того, деякі організації використовують комбінацію цих інструментів для задоволення своїх конкретних потреб у розповсюдженні даних.

1.2 Класифікація та аналіз існуючих методів для розподілу даних у задачах прийняття рішень

Для розподілу даних у задачах прийняття рішень існує безліч методів та алгоритмів [2, 7]. Всі їх можна класифікувати за кількома категоріями, серед яких можна виділити наступні:

- методи випадкової вибірки;
- методи повторної вибірки;
- кластерні методи;
- методи поділу даних;
- методи зменшення розмірності;
- алгоритми зіставлення;
- алгоритми дерева рішень.

1.2.1 Методи випадкової вибірки

Методи випадкової вибірки передбачають випадковий вибір точок даних із набору даних, що допомагає створити репрезентативні підмножини для завдань аналізу чи прийняття рішень [2]. Ці методи особливо корисні, коли необхідно уникнути упередженості та переконатися, що кожна точка даних має однакові шанси бути включеною до вибірки. До цієї категорії відносяться:

– проста випадкова вибірка. У простій випадковій вибірці кожна точка даних у наборі даних має однакову ймовірність бути обраною, що робить її абсолютно випадковою;

– стратифікована вибірка. Стратифікована вибірка передбачає розподіл даних на взаємовиключні підгрупи, або страти, на основі конкретних критеріїв, а потім вибірку з кожної страти незалежно;

– систематичний відбір. Систематична вибірка передбачає вибір точок даних через регулярні проміжки часу з відсортованого набору даних;

– кластерна вибірка. Під час кластерної вибірки сукупність розподіляється на кластери або групи, а потім випадковим чином обираються цілі кластери для вибірки, включаючи всіх осіб у цих кластерах [8];

– багатоетапна вибірка. Багатоетапна вибірка – це комбінація різних методів вибірки. Він передбачає кілька етапів вибірки, починаючи з великих кластерів і поступово переходячи до менших підгруп у кожному кластері;

– вибірка ймовірності, пропорційної розміру. Вибірка PPS відбирає точки даних з імовірністю, прямо пропорційною їхньому розміру або вазі в сукупності.

Методи випадкової вибірки мають ряд переваг і недоліків. Серед переваг можна виділити:

– репрезентативність. Методи випадкової вибірки призначені для забезпечення репрезентативної вибірки з більшої сукупності, зменшуючи ризик упередженості. Кожен елемент генеральної сукупності має рівні шанси потрапити до вибірки;

– статистична достовірність. Зразки, отримані шляхом випадкової вибірки, часто є статистично достовірними, що полегшує узагальнення результатів для всієї сукупності [3];

– простота. Методи випадкової вибірки відносно прості у застосуванні, що робить їх доступними для дослідників та організацій з обмеженими ресурсами;

– неупередженість. Випадкова вибірка є неупередженою, що означає, що дослідники навмисно не віддають перевагу певним групам чи елементам із вибірки чи не виключають їх із неї;

– легкість оцінки. Статистичні властивості вибірки, такі як межа похибки, можна оцінити більш точно, якщо використовується випадкова вибірка;

– можливість узагальнення. Результати випадково відібраної вибірки часто можна узагальнити на ширшу сукупність, що має вирішальне значення для отримання значущих висновків.

До недоліків можна віднести:

– ресурсовитратні. У деяких випадках випадкові методи вибірки можуть бути ресурсомісткими, особливо коли популяція велика, або коли потрібна стратифікація;

– недопредставленість. Методи випадкової вибірки не гарантують представлення конкретних підгруп. Невеликі підгрупи в сукупності можуть бути недостатньо представлені у вибірці;

– складність у великих популяціях. Для дуже великих популяцій може бути важко створити справжню випадкову вибірку. У таких випадках дослідники можуть покладатися на наближення;

– неефективність. У деяких ситуаціях методи випадкової вибірки можуть бути неефективними, оскільки вони можуть призвести до включення елементів, які надають обмежену інформацію для цілей дослідження;

– помилка вибірки. Хоча випадкова вибірка мінімізує зміщення, вона не усуває можливість помилки вибірки, яка може вплинути на точність оцінок;

– проблеми зі структурою вибірки. Якщо основа вибірки (список елементів, з яких складається вибірка) неповна, застаріла або не є репрезентативною для генеральної сукупності, випадкова вибірка може не дати справді репрезентативної вибірки [9];

– складність для рідкісних подій. Під час вивчення рідкісних подій або умов випадкова вибірка може не дати достатньої кількості випадків для змістовного аналізу;

– відсутність контролю. Дослідники мають обмежений контроль над складом вибірки, що може бути недоліком, коли інтерес становлять конкретні підгрупи;

– час і вартість. Проведення опитувань із випадковою вибіркою або збір даних може зайняти багато часу та коштувати, особливо коли дослідники прагнуть отримати високу кількість відповідей.

Таким чином, методи випадкової вибірки забезпечують об’єктивні та репрезентативні вибірки для аналізу. Однак вони можуть мати обмеження, особливо коли йдеться про забезпечення представництва конкретних підгруп або у випадках, коли населення є великим і різноманітним.

1.2.2 Методи повторної вибірки

Методи повторної вибірки необхідні для роботи з великими наборами даних, обробки незбалансованих даних, оцінки статистики та оцінки продуктивності моделі [5]. До цих методів відносяться:

– SMOTE. Це техніка повторної вибірки, яка використовується для усунення дисбалансу класів у машинному навчанні. Метод генерує синтетичні зразки для класу меншості шляхом інтерполяції між існуючими зразками;

– перехресна перевірка. Перехресна перевірка – це техніка повторної вибірки, яка використовується для оцінки продуктивності прогнозних моделей. Метод передбачає поділ даних на кілька підмножин для навчання та тестування, що допомагає оцінити стабільність і узагальнення моделі;

– повторна вибірка методом ножа. Повторна вибірка за принципом ножа – це техніка для оцінки зміщення та дисперсії статистичного оцінювача шляхом повторного пропускання однієї точки даних за раз для обчислення кількох оцінок.

Переваги методів повторного відбору:

– зменшена складність даних. Ці методи дозволяють аналізувати менший, більш керований набір даних;

– покращена продуктивність моделі. Повторна вибірка може усунути незбалансовані дані, створюючи більш точні моделі;

– краще оцінювання. Методи повторної вибірки допомагають оцінити статистичні властивості та оцінити невизначеність оцінок;

– перевірка моделі. Перехресна перевірка допомагає оцінити надійність і продуктивність прогнозних моделей.

Недоліки методів повторної вибірки:

– втрата даних. Методи повторної вибірки можуть призвести до втрати деяких даних, що потенційно може вплинути на якість аналізу;

– збільшення обчислювальних витрат. Деякі методи, як-от початкове завантаження або *k*-кратна перехресна перевірка, можуть потребувати інтенсивних обчислень;

– потенційне зміщення. Неналежне використання методів повторної вибірки може внести зміщення в результати аналізу або моделювання.

1.2.3 Кластерні методи

Кластерні методи спрямовані на розділення набору даних на кластери, де точки даних у межах одного кластера більш схожі одна на одну, ніж на точки в інших кластерах [9-12]. Мета полягає в тому, щоб виявити властиві групування або закономірності в даних. Ці методи часто використовуються в аналізі даних для виявлення закономірностей, структур і зв’язків у даних.

До цих методів відносяться:

– метод *k*-середніх [10]. *K*-means є одним із найпоширеніших алгоритмів кластеризації. Він розбиває дані на *k* кластерів, де *k* є параметром, визначеним користувачем. Точки даних призначаються кластерам таким чином, щоб мінімізувати суму квадратів відстаней між точками даних і центроїдом призначеного їм кластера. Цільовою функцією для алгоритму кластеризації *K*-середніх є функція квадрата помилки:

$$J = \sum_{i=1}^{k} \sum_{j=1}^{n} (\|x_i - v_j\|)^2 = 1, \quad (1.1)$$

де k – кількість точок кластеру;

n – кількість кластерів;

x_i – точка кластеру;

v_j – центроїд;

$\|x_i - v_j\|$ – евклідова відстань між точкою x_i і центроїдом v_j;

– ієрархічна кластеризація. Ієрархічна кластеризація створює деревоподібну структуру (дендрограму) кластерів, де листи представляють окремі точки даних, а гілки представляють злиття кластерів на різних рівнях. Для певної кластеризації можна вибрати необхідний рівень деталізації;

– DBSCAN. DBSCAN – це алгоритм кластеризації на основі щільності. Він ідентифікує кластери як області високої щільності точок даних, розділені областями меншої щільності. Точки даних, які не є частиною жодного кластера, вважаються шумом;

– моделі суміші Гауса [12]. GMM – це ймовірнісна модель, яка представляє дані як суміш кількох розподілів Гауса. Метод припускає, що дані генеруються з комбінації цих розподілів і призначає точки даних кластерам на основі ймовірності приналежності до кожного компонента Гауса;

– агломеративна кластеризація. Агломеративна кластеризація починається з окремих точок даних як окремих кластерів і періодично об’єднує найближчі кластери, доки не буде виконано умову завершення. Це призводить до ієрархії кластерів;

– Fuzzy *C*-Means. FCM є розширенням *K*-means, яке дозволяє точкам даних належати до кількох кластерів із різним ступенем членства. Він призначає точки даних для кластерів на основі їх подібності до центроїдів кластера [13, 14].

Переваги кластерних методів:

– виявлення шаблонів. Методи на основі кластерів допомагають виявити приховані шаблони та структури в даних, що дозволяє приймати обґрунтовані рішення;

– групування подібних даних. Групуючи подібні точки даних, ці методи можуть спростити задачі прийняття рішень;

– покращене розуміння. Кластери можуть забезпечити більш інтуїтивне розуміння даних, дозволяючи особам, які приймають рішення, зосередитися на конкретних підмножинах;

– цільові втручання. Підприємства та організації можуть адаптувати свої стратегії до конкретних кластерів або сегментів, підвищуючи ефективність і результативність.

Недоліки кластерних методів:

– залежність від параметрів. Ефективність кластерних методів може залежати від визначених користувачем параметрів, які можуть вимагати тонкого налаштування;

– чутливість до попередньої обробки даних. На якість результатів кластера може вплинути якість попередньої обробки даних і вибір міри відстані або подібності;

– складність і обчислення. Деякі методи, наприклад ієрархічна кластеризація, можуть потребувати інтенсивних обчислень і не підходять для великих наборів даних;

– обмежено евклідовими даними. Деякі методи кластеризації припускають метрику евклідової відстані, яка може не підходити для всіх типів даних.

1.2.4 Методи поділу даних

Методи поділу даних передбачають поділ набору даних на підмножини або розділи на основі конкретних критеріїв або атрибутів. Ці методи допомагають упорядковувати дані для ефективного аналізу та прийняття рішень [2]. Вони широко використовуються в різних областях, включаючи бази даних, сховища даних і розподілені обчислення.

До методів поділу даних належать:

– поділ діапазону. Поділ діапазону передбачає розподіл даних на інтервали або діапазони на основі певного значення атрибута. Кожен діапазон може представляти певну категорію або клас даних;

– поділ на основі хешу. Розбиття на основі хешування розподіляє дані на розділи на основі хеш-функції, застосованої до певного атрибута. Це забезпечує відносно рівномірний розподіл даних між розділами;

– циклічне розділення. Циклічне розподілення циклічно розподіляє дані між різними блоками прийняття рішень або місцями зберігання, гарантуючи, що кожен блок отримує справедливу частку даних;

– поділ списку. Розділення списків передбачає вручну вказувати, які елементи даних належать до кожного розділу, зазвичай на основі попередньо визначених списків або категорій;

– поділ на основі дат. Поділ на основі дати розподіляє дані на основі певних дат або інтервалів часу, що робить його придатним для даних часових рядів;

– поділ на основі значень. Розбивка на основі значень групує дані на основі визначеного значення або порогу, часто використовується в сценаріях, де релевантним є числовий атрибут.

Серед переваг методів поділу даних можна виділити:

– ефективне надсилання запитів. Розділення даних може значно підвищити продуктивність запитів, зменшивши обсяг даних, які потрібно сканувати для конкретного запиту;

– паралельна обробка. Розділені дані можна обробляти паралельно, що сприяє швидшому прийняттю рішень і аналізу;

– покращена організація даних. Розділення робить організацію даних більш інтуїтивно зрозумілою, полегшуючи розуміння та роботу з даними особам, які приймають рішення;

– покращена безпека даних. Конфіденційні дані можна розділити на розділи з різними рівнями контролю доступу, допомагаючи забезпечити безпеку даних і відповідність.

Недоліки методів розподілу даних:

– перекос даних. Нерівномірний розподіл даних між розділами може призвести до перекосу даних і нерівномірного навантаження на обробку;

– складність. Створення розділів і керування ними може викликати складність, особливо в розподілених системах або великих базах даних;

– накладні витрати на перерозподіл. Коли потреби в розподілі даних змінюються, перерозподіл може потребувати ресурсів;

– втрата глобального аналізу. У деяких випадках розділення може обмежити можливість виконувати глобальний аналіз усього набору даних.

1.2.5 Методи зменшення розмірності

Методи зменшення розмірності – це методи, які використовуються для зменшення кількості функцій або атрибутів у наборі даних, зберігаючи якомога більше релевантної інформації. Ці методи є цінними для розподілу даних в задачах прийняття рішень, оскільки вони допомагають спростити складні набори даних, роблячи їх більш зручними для аналізу та інтерпретації. Вони знаходять застосування в таких сферах, як: фінанси, охорона здоров’я, обробка природної мови та аналіз зображень [15-17].

До методів зменшення розмірності відносяться:

– аналіз основних компонентів. PCA – це лінійний метод зменшення розмірності, який перетворює вихідні функції в новий набір некорельованих змінних, які називаються головними компонентами. Ці компоненти фіксують максимальну дисперсію в даних;

– *t*-SNE. Метод *t*-SNE – це нелінійна техніка зменшення розмірності, яка спрямована на візуалізацію даних великої розмірності в нижчих розмірах, зберігаючи при цьому зв’язки подібності між точками даних;

– автокодери. Автокодери – це тип штучної нейронної мережі, яка використовується для неконтрольованого навчання. Вони складаються з кодера та декодера, і вчаться стискати вхідні дані в представлення нижчої розмірності, а потім реконструювати вихідні дані;

– лінійний дискримінантний аналіз. LDA – це контрольований метод зменшення розмірності, який максимізує поділ між класами, одночасно зменшуючи розмірність даних. Його часто використовують у задачах класифікації;

– факторний аналіз. Факторний аналіз – це статистичний метод, який використовується для виявлення прихованих факторів, які пояснюють спостережувані кореляції між змінними. Він зменшує розмірність, замінюючи спостережувані змінні меншою кількістю прихованих факторів.

Переваги методів зменшення розмірності:

– спрощені дані. Зменшена розмірність робить дані легшими для керування та легшими для візуалізації та інтерпретації;

– зменшення шуму. Зменшення розмірності може допомогти видалити шум і нерелевантну інформацію з даних, покращуючи процес прийняття рішень;

– обчислювальна ефективність. Менші набори даних, отримані в результаті зменшення розмірності, потребують менше обчислювальних ресурсів для аналізу [16];

– краще узагальнення. Зменшена розмірність може призвести до моделей, які краще узагальнюють, особливо в задачах машинного навчання та прогнозування.

Недоліки методів зменшення розмірності:

– втрата інформації. Методи зменшення розмірності можуть призвести до втрати інформації, що може вплинути на точність прийняття рішень;

– складність. Деякі методи, як-от глибокі автокодери, можуть бути складними для впровадження та потребують великих обчислювальних ресурсів;

– налаштування параметрів. Вибір правильних параметрів для методів зменшення розмірності може бути складним завданням.

1.2.6 Алгоритми зіставлення

Алгоритми зіставлення – це методи, які використовуються для вирішення проблем, у яких пари елементів потрібно зіставити таким чином, щоб відповідати певним критеріям або обмеженням. Ці алгоритми широко застосовуються в різних сферах для оптимізації розподілу ресурсів, прийняття рішень і об’єднання осіб або організацій на основі переваг або сумісності.

Серед алгоритмів зіставлення можна виділити:

– зіставлення максимальної потужності. Алгоритм зіставлення максимальної потужності мають на меті знайти найбільший набір пар, що не перекриваються, на графі, де ребра представляють потенційні пари;

– зіставлення дводольного графа. Зіставлення дводольного графа використовується для об’єднання елементів із двох різних груп на основі переваг, обмежень або сумісності, і часто передбачає створення дводольного графа, де ребра представляють потенційні збіги;

– алгоритм Гейла-Шеплі. Алгоритм Гейла-Шеплі, також відомий як проблема стабільного шлюбу, використовується для пошуку стабільної відповідності між двома групами елементів, наприклад чоловіками та жінками, де кожен елемент має ранжовані переваги над елементами в іншій групі [1, 17-21].

Переваги алгоритмів зіставлення:

– оптимізація. Алгоритми зіставлення можуть оптимізувати розподіл ресурсів, прийняття рішень і об’єднання, що призводить до ефективних рішень;

– чесність. Ці алгоритми гарантують, що збіги створюються на основі вподобань і обмежень, сприяючи справедливості та сумісності;

– масштабованість. Алгоритми відповідності можуть вирішувати масштабні проблеми, що робить їх придатними для складних завдань прийняття рішень.

Недоліки алгоритмів зіставлення:

– складність. Деякі проблеми зіставлення можуть бути обчислювально складними, вимагаючи значних обчислювальних ресурсів;

– відсутність глобального оптимуму. У деяких випадках алгоритми зіставлення можуть знайти не глобальний оптимум, а локально оптимальне рішення.

Алгоритми зіставлення можуть полегшити розподіл ресурсів, прийняття рішень і створення пар у різних сферах, від охорони здоров’я та транспорту до онлайн-знайомств і призначення на роботу [21, 22].

1.2.7 Алгоритми дерева рішень

Алгоритми дерева рішень – це сімейство алгоритмів машинного навчання, які використовуються як для завдань класифікації, так і для регресії.

Дерева рішень є візуальним представленням процесу прийняття рішень і можуть застосовуватися в різних областях для прийняття рішень на основі вхідних характеристик. Вони особливо корисні для розповсюдження даних для прийняття рішень, оскільки вони надають чіткий спосіб, який можна інтерпретувати, для планування рішень.

Прикладами алгоритмів дерева рішень є:

– базове дерево рішень. Базове дерево рішень складається з вузлів, що представляють характеристики, гілок, що представляють рішення або результати, і листів, що представляють остаточне рішення або класифікацію. Його створено за допомогою рекурсивного процесу вибору найкращих функцій для розділення даних;

– CART – це алгоритм дерева рішень, який можна використовувати як для завдань класифікації, так і для регресії [23]. Він працює шляхом рекурсивного розбиття набору даних на підмножини на основі найбільш інформативних характеристик. Для цього використовується критерій Gini Index:

$$Gini = 1 - \sum_{i=1}^{c} P_i^2, \tag{1.2}$$

де c – кількість класів;

P – ймовірність віднесення об’єкта до певного класу;

– ID3 – це алгоритм дерева рішень, який в основному використовується для завдань класифікації. Він вибирає функції на основі отримання інформації та будує дерево шляхом рекурсивного поділу даних на підмножини [8, 24];

– C4.5 є розширенням ID3 і використовується для класифікаційних завдань. Він використовує статистичний підхід для оцінки важливості ознак і створення дерев рішень;

– випадковий ліс. Випадковий ліс – це метод ансамблевого навчання, який поєднує кілька дерев рішень для підвищення точності та зменшення надмірного оснащення. Він створює ліс дерев рішень і агрегує їхні прогнози;

– дерева посилення градієнта. Дерева посилення градієнта, такі як XGBoost і LightGBM, є методами ансамблю, які послідовно створюють дерева рішень, при цьому кожне дерево виправляє помилки попередніх. Вони дуже ефективні та популярні як для завдань класифікації, так і для регресії.

Переваги алгоритмів дерева рішень:

– здатність тлумачення. Дерева рішень добре інтерпретуються та можуть використовуватися для пояснення міркувань, що стоять за рішеннями;

– гнучкість. Вони можуть обробляти як категоричні, так і числові дані та є універсальними для завдань класифікації та регресії;

– простота використання. Дерева рішень відносно легко зрозуміти, реалізувати та візуалізувати;

– без припущень. Вони не роблять серйозних припущень щодо розподілу даних.

Недоліки алгоритмів дерева рішень:

– переобладнання. Дерева рішень можуть переобладнати навчальні дані, що призводить до поганого узагальнення;

– зміщення. Вибір кореневого вузла та критеріїв поділу може внести зміщення в модель;

– нестабільність. Невеликі зміни в даних можуть призвести до істотно різних дерев;

– обмежена експресивність. Дерева рішень можуть мати проблеми зі складними зв’язками в даних.

Вибір методу або алгоритму для використання залежить від характеру проблеми прийняття рішення, наявних даних, обмежень і цілей. Часто завдання прийняття рішень у реальному світі можуть вимагати комбінації цих методів для досягнення ефективних та ефективних рішень.

1.3 Особливості задач прийняття рішень

Прийняття рішень – процедура вибору найкращої альтернативи із заданої множини на підставі певного критерію або множини критеріїв [2]. Ці проблеми можуть бути досить складними, і вони часто включають невизначеність, компроміси та численні критерії.

Задачі прийняття рішень, як правило, включають низку можливих варіантів або альтернатив. Ці альтернативи можуть відрізнятися з точки зору вартості, переваг, ризиків та інших атрибутів. Особа, яка приймає рішення, повинна вибрати найкращу альтернативу або комбінацію альтернатив [2].

Часто в задачах прийняття рішень містяться як об’єктивні, так і суб’єктивні елементи. Об’єктивні фактори, такі як вартість або технічні характеристики, можна визначити кількісно, тоді як суб’єктивні фактори, такі як особисті переваги чи нематеріальні вигоди, виміряти важче.

Особи, які приймають рішення, часто стикаються з невизначеністю щодо майбутнього, оскільки вони не можуть з упевненістю передбачити всі можливі результати. Ця невизначеність може бути спричинена такими факторами, як динаміка ринку, зміни навколишнього середовища або поведінка конкурентів.

Багато проблем прийняття рішень підлягають часовим обмеженням. Рішення потрібно приймати протягом певного періоду часу, що може вплинути на рівень аналізу та доступність даних для прийняття рішень [25].

Особи, які приймають рішення, часто стикаються з компромісами між суперечливими цілями. Наприклад, мінімізація витрат може суперечити максимізації якості або задоволенню клієнтів. Пошук правильного балансу є ключовим аспектом прийняття рішень.

Задачі прийняття рішень часто включають кілька критеріїв або цілей. Ці критерії можуть включати фінансові фактори (наприклад, вартість, дохід), нефінансові фактори (наприклад, вплив на навколишнє середовище,

соціальна відповідальність) та інші міркування (наприклад, дотримання законодавства, моральний стан працівників) [7].

В організаціях процес прийняття рішень часто є колективним, в якому беруть участь кілька людей або команд. Групова динаміка та міжособистісні стосунки можуть впливати на процес прийняття рішень.

Особи, які приймають рішення, можуть бути вразливі до когнітивних упереджень, які впливають на їхні судження та вибір. Поширені упередження включають упередження підтвердження, прив’язку та надмірну впевненість, що може призвести до неоптимальних рішень.

В епоху великих даних особи, які приймають рішення, можуть бути перевантажені величезними обсягами інформації. Сортувати цю інформацію, щоб зробити обґрунтований вибір, є складним завданням.

Деякі задачі прийняття рішень включають етичні та моральні дилеми. Рішення можуть впливати на різні зацікавлені сторони, і етичні міркування можуть відігравати важливу роль у процесі прийняття рішень.

Прийняття рішень часто є повторюваним процесом [2]. Після реалізації рішення особи, які приймають рішення, повинні контролювати його результати та вносити корективи на основі зворотного зв’язку. Ця петля зворотного зв’язку має вирішальне значення для постійного вдосконалення.

Різні люди та організації мають різні рівні толерантності до ризику. Деякі уникають ризику, віддаючи перевагу безпечному вибору, тоді як інші більш терпимі до ризику та готові йти на прорахований ризик.

Для допомоги тим, хто приймає рішення, доступні різні інструменти та методи підтримки прийняття рішень, включаючи матриці рішень, аналіз витрат і вигод, дерева рішень та моделі оптимізації [2, 7, 24].

Контекст рішення може суттєво вплинути на процес прийняття рішення. Одна й та сама проблема вирішення може вимагати різних підходів у різних контекстах.

Те, як задача прийняття рішення сформульована або представлена, може вплинути на вибір особи, яка приймає рішення.

Спосіб подання інформації може призвести до різних рішень, навіть якщо базові варіанти залишаються незмінними.

Прийняття рішень часто залежить від нормативних та правових обмежень, які необхідно враховувати. Недотримання цих обмежень може призвести до юридичних і фінансових наслідків.

Під час прийняття рішення необхідно розуміти всі ці особливості. Особи, які приймають рішення, повинні ретельно проаналізувати ситуацію, врахувати ці фактори та використовувати відповідні інструменти та стратегії, щоб зробити найкращий можливий вибір.

1.4 Аналіз літературних джерел щодо апробації результатів стосовно розподілу даних у задачах прийняття рішень

У джерелі [1] досліджується застосування математичної концепції під назвою «проблема стабільного шлюбу» до контексту вступу абітурієнтів до коледжу. В роботі представлено алгоритм Гейла-Шеплі, фундаментальний інструмент у теорії відповідності та теорії ігор, та доведено, що для будь-якої рівної кількості учасників кожного типу завжди можна знайти відповідність, у якій усі пари стабільні. Під час аналізу даного джерела обрано алгоритм для наукового дослідження.

У навчальному посібнику [2] викладено теоретичні основи прийняття рішень, описані основні поняття процесу прийняття рішень, види моделей та методів, основні технології прийняття рішень в інформаційних системах, технології оцінювання та аналізу, технології прийняття колективних рішень. У джерелі надано практичні приклади застосування технологій прийняття рішень, описано сучасний стан розвитку технологій прийняття рішень в інформаційних системах.

У джерелі [7] обговорюється вибір методів для підвищення надійності рішення та скорочення часу дослідження простору станів конкретного

об’єкта. Він визначає ключові фактори, що впливають на прийняття рішень у середовищі розробки нечітких процесів, представляючи їх природною мовою, щоб підвищити впевненість у прийнятті рішень. У роботі також пропонуються інформаційні технології для оцінки повноти цілей прийняття рішень на основі цих факторів, оцінки ефективності за допомогою нечіткого висновку та підходу Заде-Мамдані. Тестування програмного забезпечення перевіряє ці методи, підтверджуючи покращену надійність рішень у різних областях.

У джерелі [18] розглядаються практичні проблеми двостороннього зіставлення. Основна мета роботи полягає в тому, щоб розширити алгоритм ледачого Гейла-Шеплі, або Lazy Gale-Shapley, до практичного сценарію багато-до-одного, коли коледжі можуть приймати кількох студентів. В статті представлено ієрархію студентів, досліджуючи залежність її продуктивності від вибору ієрархії. Досліджуються ефективні умови для отримання оптимальної ієрархії, особливо зосереджуючись на сумісності між частковими перевагами коледжів.

Метою роботи [19] є вивчення застосування алгоритму Гейла-Шеплі та його варіацій для автоматизації вступу до навчальних закладів, зокрема в контексті вступу до вищих навчальних закладів в Україні з 2016 по 2019 роки. Дослідження зосереджено на проблемах та відхиленнях від ідеальні припущення алгоритму, такі як квоти, неповні списки пріоритетів та асиметрія інформації серед заявників. Це підкреслює потенційну втрату оптимальності та ефективності результатів розподілу через ці фактори.

Метою роботи [20] є аналіз розширеної версії запропонованого алгоритму Ірвінга для стабільних проблем відповідності. Алгоритм продовжується, доки кожен агент не отримає пропозицію або агент не буде відхилений усіма в їх списку переваг. Дослідження досліджує поведінку алгоритму за припущення незалежного рівномірного рейтингу. Результати демонструють, що з високою ймовірністю кожен агент знаходить партнера, причому як кількість агентів у циклах певної довжини, так і загальна

кількість стабільних збігів обмежені. Загальна кількість пропозицій демонструє певну асимптотику.

У статті [21] досліджено теорію відповідності та її практичне застосування в економіці. В роботі визначено ключові поняття «відповідність», «індивідуально раціональна відповідність» та «стійка відповідність», реалізовано алгоритм Гейла-Шеплі для пошуку оптимальних пар між особами в різних групах. У статті обговорюються різні прикладні сценарії для теорії відповідності, включаючи моделі фірма-працівник, ринки праці молодих лікарів, вибір коледжу та розподіл стипендій. Він також представляє блок-схему алгоритму Гейла-Шеплі для потенційної реалізації програмного забезпечення, спрямованого на підвищення практичної користі теорії відповідності в сценаріях реального світу.

Метою роботи [22] є підвищення ефективності та взаємодії з користувачами платформ диспетчеризації таксі на вимогу шляхом впровадження системи призначення завдань з урахуванням переваг. На відміну від попередньої політики, яка зосереджувалася виключно на прибутку від платформи, цей підхід враховує вподобання як водія, так і пасажира. Запропонована модель, онлайн-стабільна відповідність за відомими ідентичними незалежними розподілами, або OSM-KIID, спрямована на максимізацію загального прибутку при мінімізації кількості блокуючих пар, задовольняючи вподобання користувачів. Онлайн-алгоритм LP-ALG досягає сприятливих результатів, перевершуючи інші методи в сценаріях із більшою кількістю завдань, ніж драйверів, що підтверджено оцінюванням синтетичних і реальних наборів даних.

Метою роботи [26] є вивчення впливу соціальних мереж на багатоаспектне прийняття рішень. У статті представлено структуру, засновану на аналізі соціальних мереж, щоб досягти багатоатрибутної відповідності соціальних мереж, навіть якщо відносна вага впливових осіб невідома. Це передбачає створення матриці оцінки довіри через відносини поширення довіри, обчислення ваг впливових факторів за допомогою моделі

нелінійної оптимізації та інтеграцію цих ваг у процес оцінювання. Крім того, у роботі будуються багатоцільові моделі оптимізації для раціональних і консервативних агентів для створення оптимальних відповідних пар. Результати демонструють ефективність цього методу зіставлення соціальних мереж.

Метою роботи [27] є вирішення потреби в новій структурі зв'язку в системах наступного покоління шляхом вивчення використання висотних платформ, або HAP, і безпілотних літальних апаратів (БПЛА) у ретрансляційному зв'язку. Дослідження представляє метод узгодження на основі алгоритму Гейла-Шеплі для сполучення HAP і БПЛА. Чисельне моделювання демонструє, що GSA перевершує випадкове зіставлення, демонструючи його чудову продуктивність у підвищенні ефективності релейного зв'язку.

1.5 Постановка задачі дослідження

Таким чином, дослідження методу Гейла-Шеплі є актуальним, оскільки застосування цього алгоритму може допомогти в розв'язанні проблем адекватного розподілу обмежених ресурсів в різних сферах, таких як економіка, інформаційні технології, медицина та інші [18-22, 26]. Тому ставиться завдання розробки застосунку для розподілу волонтерів між волонтерськими проєктами із застосуванням методу Гейла-Шеплі. Останні роки значно збільшилася кількість волонтерів та проєктів, що їх потребує. А отже існує і потреба в застосунку, що допоміг би волонтерам та проєктам знаходити одне одного.

Об'єктом дослідження є процес розроблення застосунку для розподілу волонтерів між волонтерськими проєктами.

Метою дослідження є розробка застосунку для розподілу волонтерів між волонтерськими проєктами на основі методу Гейла-Шеплі, що призначений для створення стабільних відповідностей..

Для досягнення мети необхідно вирішити такі завдання:

– провести аналіз існуючих методів розподілу даних у задачах прийняття рішень;

– провести аналіз методу Гейла-Шеплі;

– сформувати бізнес-логіку застосунку;

– реалізувати алгоритм Гейла-Шеплі;

– розробити та реалізувати програмний застосунок для розподілу волонтерів між волонтерськими проєктами, в основі якого застосовано метод Гейла-Шеплі;

– протестувати розроблений застосунок.

2 ОСОБЛИВОСТІ МЕТОДУ ГЕЙЛА-ШЕПЛІ ДЛЯ РОЗПОДІЛУ ДАНИХ У ЗАДАЧАХ ПРИЙНЯТТЯ РІШЕНЬ

2.1 Особливості методу Гейла-Шеплі

Проблема створення стабільної відповідності між двома наборами елементів з різними перевагами для кожного з них – це проблема, з якою суспільство стикається ледь не щодня. Дана проблематика відноситься до теорії ігор, що є розгалуженням штучного інтелекту. Прикладом є розподіл вакантних робочих місць серед кандидатів, пошук кращої відповідності між орендарем та орендодавцем, розподіл медичних спеціалістів між закладами охорони здоров'я тощо. У кожній з цих задач є два набори сутностей зі своїми наборами пріоритетів з сутностей протилежного набору.

Рішення проблеми стабільних відповідностей у 1962 році запропонували Девід Гейл і Ллойд Шеплі. У роботі [1] автори досліджують застосування математичної концепції під назвою «проблема стабільного шлюбу» до контексту вступу до коледжу.

Проблема стабільного шлюбу – це класична алгоритмічна задача, яка прагне знайти стабільну відповідність між двома наборами індивідів, наприклад, чоловіками та жінками. Гейл та Шеплі вводять поняття «стабільності» у контексті створення подружжя: набір шлюбів є нестабільним, якщо серед них є чоловік і жінка, які не одружені один з одним, але віддають перевагу один одному своїм справжнім партнерам. Автори представили алгоритм Гейла-Шеплі, який є ефективним способом знайти стабільну відповідність між двома наборами агентів із перевагами, та довели, що для будь-якої рівної кількості учасників кожного типу завжди можна знайти відповідність, у якій усі пари стабільні [1].

На рисунку 2.1 [28] зображена схема роботи алгоритму Гейла-Шеплі на прикладі задачі створення стабільних пар між студентом та проєктом.

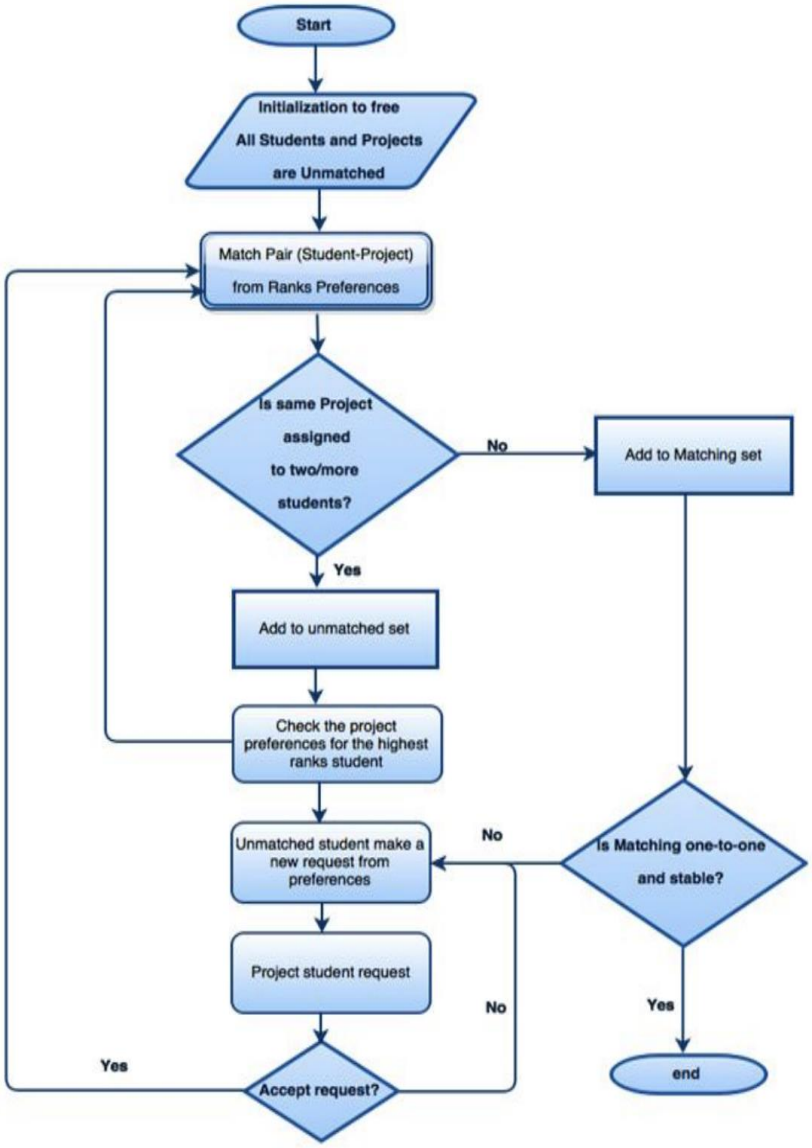

Рисунок 2.1 – Схема роботи алгоритму Гейла-Шеплі на прикладі задачі створення стабільних пар між студентом та проєктом

2.1.1 Переваги та недоліки алгоритму Гейла-Шеплі

Алгоритм Гейла-Шеплі має ряд переваг:

– гарантована стабільна відповідність;

– справедливість. Алгоритм розроблено таким чином, щоб бути справедливим, гарантуючи, що кожна особа отримає найкращого партнера, якого вона може отримати відповідно до своїх уподобань;

– простота. Алгоритм Гейла-Шеплі відносно простий для розуміння та впровадження, що робить його практичним вибором для вирішення проблеми стабільного шлюбу;

– ефективність. У більшості випадків алгоритм швидко сходиться і не потребує оцінки всіх можливих пар. Він припиняється, коли досягається стабільна відповідність, що часто трапляється через розумний проміжок часу, особливо коли переваги не надто складні.

Проте, є й певні недоліки:

– упереджені результати. Алгоритм Гейла-Шеплі за своєю суттю має упередження до однієї сторони зіставлення, як правило, до сторони пропозиції (чоловіки в класичній версії). Сторона, яка робить пропозиції, як правило, закінчується більш сприятливими результатами порівняно з іншою стороною. Це упередження можна розглядати як недолік, якщо справедливість є пріоритетом [19, 20];

– складність. Незважаючи на те, що алгоритм концептуально простий, його реалізація за допомогою великої кількості людей може стати дорогим з точки зору обчислень. Найгірша часова складність – $O(n^2)$, що може бути непрактичним для дуже великих наборів даних [18];

– відсутність гнучкості. Алгоритм Гейла-Шеплі припускає, що переваги є строгими та повними, тобто кожен індивід має чіткий порядок переваг над усіма членами протилежної групи. У реальних сценаріях уподобання можуть бути більш тонкими, і їх нелегко вловити таким чином.

Алгоритм Гейла-Шеплі є важливим внеском в вирішення проблеми стабільних відповідностей. Він знайшов практичне застосування в вирішенні життєвих задач, що стосуються ринку вакансій та безробітних, ринку донорів та акцепторів, прийому абітурієнтів в університети тощо. Алгоритм має як переваги, так і деякі недоліки. Ці міркування слід брати до уваги, вибираючи використання цього алгоритму в певних застосунках.

2.1.2 Модифікації алгоритму Гейла-Шеплі

Хоча основний алгоритм Гейла-Шеплі залишається незмінним, протягом багатьох років було запропоновано кілька модифікацій і розширень для вирішення різних сценаріїв і вимог. Ось деякі з них:

– множинні зв’язки [18]. В оригінальному алгоритмі кожен учасник унікально ранжує всі члени протилежного набору. У деяких випадках учасники можуть мати однакові переваги для кількох учасників. У таких випадках можна змінити алгоритм, щоб дозволити зв’язки, коли кілька членів протилежного набору можуть мати однаковий рейтинг. Це можна зробити, розширивши списки переваг, щоб включити кількох осіб з однаковим рангом;

– неповні вподобання. У деяких сценаріях реального світу учасники можуть не мати повних списків уподобань, тобто вони можуть не ранжувати кожного члена протилежного набору. Можна змінити алгоритм для обробки неповних списків уподобань, дозволивши учасникам подавати часткові рейтинги;

– проблема стабільних сусідів по кімнаті [19]. Оригінальний алгоритм Гейла-Шеплі передбачає дводольний граф із двома різними наборами (чоловіки та жінки). Проблема стабільних сусідів по кімнаті є різновидом проблеми стабільного підбору, де є лише один набір учасників, і кожен учасник ранжує всіх інших у порядку переваги. Цю проблему можна вирішити за допомогою модифікованої версії алгоритму Гейла-Шеплі, розробленої для цього конкретного сценарію;

– динамічні налаштування. У деяких ситуаціях уподобання учасників можуть змінюватися з часом. Алгоритм може бути адаптованим для обробки динамічних списків уподобань, де учасники можуть оновлювати свої рейтинги та повторно обчислювати стабільні збіги за потреби;

– відповідність з обмеженнями. Розширення алгоритму можуть включати додаткові обмеження або вимоги, такі як квоти, обмеження ємності або обмеження щодо кількості збігів, які може мати учасник.

Ці модифікації та розширення розроблені, щоб зробити алгоритм Гейла-Шеплі більш гнучким і застосовним до широкого діапазону сценаріїв, окрім його початкового формулювання. Залежно від конкретної проблеми може знадобитися відповідним чином налаштований алгоритм.

2.2 Механізм розподілу даних у задачах прийняття рішень

2.2.1 Збір даних

Першим кроком у процесі розподілу даних для задач прийняття рішень є збір даних [2, 7]. Цей етап передбачає збір відповідної інформації з різних джерел, щоб забезпечити основу для аналізу та прийняття обґрунтованих рішень.

Перед збором даних важливо чітко визначити вимоги до даних для конкретного завдання прийняття рішень. Це передбачає визначення типу необхідних даних, їх формату, обсягу і цілей збору даних. Розуміння того, що потрібно знати, керуватиме процесом збору даних.

Після визначення вимог до даних необхідно визначити джерела, з яких можна отримати дані. Джерела даних можуть дуже відрізнятися, і вони можуть включати:

– внутрішні дані – дані, створені або збережені у певній організації, такі як: записи про продажі, дані клієнтів, дані інвентаризації або операційні журнали;

– зовнішні дані – дані із зовнішніх джерел, які можуть включати загальнодоступні набори даних, звіти про дослідження ринку, державні бази даних або дані від сторонніх постачальників;

– дані датчиків [29-31]. У певних контекстах дані, зібрані з датчиків, пристроїв Інтернету речей або систем моніторингу, можуть бути цінними джерелами інформації;

– опитування та анкети. Необхідно провести опитування або зібрати інформацію за допомогою анкет, щоб отримати конкретні дані безпосередньо від окремих осіб або організацій;

– вебзбирання. Для вебданих можна використовувати методи вебзбирання для отримання інформації з вебсайтів або онлайнових баз даних;

– API. Інтерфейси прикладного програмування можуть надавати доступ до структурованих даних з онлайн-сервісів, платформ соціальних мереж або інших вебплатформ [32].

Залежно від джерел даних, для збору даних можуть використовуватися різні методи:

– автоматичний збір даних. Для цифрових джерел даних, таких як бази даних, API або дані датчиків, автоматизовані методи часто використовуються для збору даних через регулярні проміжки часу. Це може передбачати написання сценаріїв або використання інструментів інтеграції даних для отримання та зберігання даних;

– збір даних вручну. У випадках, коли дані неможливо отримати автоматично, можуть знадобитися методи збору даних вручну. Це може включати ручне введення даних із паперових записів або проведення інтерв’ю та опитувань;

– придбання даних. У деяких випадках організації можуть вибрати придбання даних у сторонніх постачальників або служб передплати, особливо коли потрібні спеціальні дані.

Після збору дані мають зберігатися безпечним і організованим способом. Це може включати використання баз даних, сховищ даних або хмарних рішень для зберігання. Належне зберігання даних полегшує пошук і обмін на етапі розповсюдження.

Ефективний збір даних закладає основу для наступних етапів процесу прийняття рішень, включаючи аналіз даних, розробку моделі та прийняття обґрунтованих рішень. Це критично важливий крок, який вимагає ретельного планування та виконання, щоб гарантувати, що зібрані дані мають високу якість і відповідають цілям завдання прийняття рішень.

2.2.2 Попередня обробка даних

Другим кроком у процесі розподілу даних для завдань прийняття рішень є попередня обробка даних. Попередня обробка даних передбачає підготовку та очищення зібраних даних, щоб забезпечити їх високу якість, стандартизованість і придатність для аналізу чи поширення [33].

Необхідно визначити та виправити відсутні точки даних. Загальні методи включають імпутацію (заміну відсутніх значень розрахунковими значеннями), видалення рядків або стовпців із надмірною кількістю відсутніх даних або використання спеціалізованих алгоритмів для імпутації відсутніх даних.

Також необхідно виявити та обробити викиди, які можуть спотворити результати аналізу чи моделювання. Можна видалити викиди, трансформувати їх або обмежити крайні значення [34].

Якщо дані було зібрано з кількох джерел, то за потреби можна об’єднати їх в один набір даних. Необхідно переконатися, що дані з різних джерел сумісні та можуть бути ефективно об’єднані.

Важливим є нормалізація даних та масштабування числових функцій, щоб привести їх до загального масштабу. Такі методи, як min-max масштабування або стандартизація (нормалізація *z*-показника), можуть бути застосовані, щоб гарантувати, що функції мають однакові діапазони та одиниці [35, 36].

Формула min-max масштабування:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}. \quad (2.1)$$

Формула стандартизації або нормалізації z-показника:

$$x' = \frac{x - \bar{x}}{\sigma}, \quad (2.2)$$

де $\bar{x}$ – середнє значення x;

σ – стандартне відхилення x.

У випадках, коли набір даних має велику кількість функцій, можна застосувати методи зменшення розмірності, такі як метод головних компонент або методи вибору ознак, щоб зменшити кількість змінних, зберігаючи важливу інформацію.

Необхідно забезпечити узгодженості даних, переконатися, що одиниці даних, формати та структури узгоджені в усьому наборі даних [35]. Стандартизація допомагає точно порівнювати та аналізувати дані.

Попередня обробка даних є критично важливим етапом у процесі розповсюдження даних, оскільки вона суттєво впливає на точність і надійність подальшого аналізу та завдань прийняття рішень. Добре попередньо оброблені дані, швидше за все, дадуть значущу інформацію та підтримають обґрунтовані рішення.

2.2.3 Аналіз даних

Етап аналізу даних у процесі розповсюдження даних для завдань прийняття рішень є критичним етапом, на якому дані досліджуються та

аналізуються, щоб отримати розуміння, виявити закономірності та отримати значущу інформацію. Для цього використовуються різні техніки і методи.

Аналіз даних часто починається з попередньої обробки та очищення даних, як згадувалося раніше. Цей крок має вирішальне значення для того, щоб дані були точними, повними та придатними для аналізу.

Одним з видів аналізу є описовий аналіз. Метою описового аналізу є узагальнення та опис основних характеристик даних. Загальні методи включають обчислення підсумкових статистичних даних (наприклад, середнього значення, медіани, стандартного відхилення), створення частотних розподілів і створення візуалізацій даних (наприклад, гістограм, прямокутних діаграм, стовпчастих діаграм), щоб отримати початкове розуміння розподілу даних і закономірностей.

Також використовується дослідницький аналіз даних. EDA – це підхід до аналізу даних, який зосереджується на візуальному дослідженні та відкритті. Аналітики використовують EDA для визначення тенденцій, викидів, зв’язків між змінними та потенційних сфер інтересу. Візуалізація даних є центральним компонентом EDA, який допомагає аналітикам виявляти приховані закономірності в даних.

Інференційна статистика передбачає створення висновків або прогнозів щодо сукупності на основі вибірки даних. Це включає перевірку гіпотез, довірчі інтервали та регресійний аналіз. Інференційна статистика допомагає аналітикам робити висновки та робити прогнози поза межами даних спостереження.

Для завдань прогнозного моделювання та класифікації використовуються методи машинного навчання. Ці методи включають контрольоване навчання (наприклад, лінійна регресія, дерева рішень, нейронні мережі), неконтрольоване навчання (наприклад, кластеризація, зменшення розмірності) та навчання з підкріпленням. Машинне навчання можна використовувати для таких завдань, як прогнозування поведінки клієнтів, розпізнавання зображень і обробка природної мови [29, 37, 38].

Аналіз часових рядів зосереджується на даних, зібраних за певний проміжок часу, таких як ціни на акції, погодні дані або економічні показники. Методи в цій галузі включають декомпозицію часових рядів, моделювання авторегресійної інтегрованої ковзної середньої і сезонну декомпозицію.

Аналіз тексту передбачає вилучення інформації з неструктурованих текстових даних. Він охоплює такі завдання, як аналіз настроїв, моделювання теми та обробка природної мови. Аналіз тексту зазвичай використовується для аналізу відгуків клієнтів, відстеження настроїв у соціальних мережах і категоризації вмісту [39].

Просторовий аналіз має справу з географічними даними або даними на основі розташування. Географічні інформаційні системи і просторова статистика допомагають аналізувати та візуалізувати дані, пов’язані з місцями, наприклад, наносити на карту спалахи захворювань, аналізувати транспортні мережі або оптимізувати розташування роздрібних магазинів.

Аналітики використовують різноманітне програмне забезпечення та інструменти для проведення аналізу даних [12, 15, 16]. Серед поширених варіантів – R, Python (з такими бібліотеками, як Pandas, NumPy і Scikit-Learn), SAS, SPSS і Excel. Кожен інструмент пропонує ряд функцій і можливостей для різних типів аналізу. Для оцінки продуктивності моделі та її узагальнення у машинному навчанні та прогнозному моделюванні використовують такі методи перевірки, як перехресна перевірка. Це гарантує, що моделі роблять точні прогнози на нових, ще невідомих даних.

Аналіз даних часто є ітераційним процесом. Може знадобитися переглянути попередні етапи, уточнити моделі або вивчити додаткові джерела даних на основі початкових висновків і відгуків зацікавлених сторін.

На етапі аналізу даних необроблені дані перетворюються на практичні висновки, забезпечуючи основу для прийняття обґрунтованих рішень. Важливо використовувати методи аналізу даних, які відповідають їхнім конкретним цілям і типам даних, щоб приймати ефективні та обґрунтовані рішення для складні проблеми.

2.2.4 Припущення щодо розподілу та розробка моделі розподілу даних

Припущення щодо розподілу та етап розробки моделі є важливим етапом у процесі розподілу даних для завдань прийняття рішень, особливо коли задіяні статистичні моделі або моделі машинного навчання [4, 23]. Цей крок передбачає створення припущень щодо розподілу даних і розробку моделей, які найкраще відповідають даним для прогнозування, класифікації чи висновку.

Серед припущень щодо розподілу можна виділити наступні:

– припущення нормальності. Багато статистичних методів припускають, що дані мають нормальний розподіл. Це припущення часто перевіряється за допомогою таких методів, як тест Шапіро-Вілка або візуальний огляд гістограм і графіків Q-Q. Відхилення від нормальності можуть вимагати перетворення даних або використання непараметричних методів;

– припущення незалежності. Деякі моделі, зокрема в аналізі часових рядів, припускають, що точки даних є незалежними одна від одної. Це означає, що значення однієї точки даних не залежить від значень попередніх або майбутніх точок даних. Порушення цього припущення може вимагати використання більш складних моделей;

– припущення гомоскедастичності. Гомоскедастичність припускає, що дисперсія помилок (залишків) є постійною на всіх рівнях незалежних змінних. Гетероскедастичність, коли дисперсія змінюється залежно від предикторів, може вимагати коригування або трансформації моделі;

– припущення лінійності. Моделі лінійної регресії припускають лінійний зв’язок між залежною та незалежною змінними. Якщо це припущення порушується, моделі нелінійної регресії або поліноміальна регресія можуть бути більш прийнятними.

У процесі розробки моделі можна виділити наступні етапи:

– вибір типів моделей. Виходячи з припущень щодо розподілу та характеру даних, необхідно обрати відповідні типи моделей. Загальні варіанти включають лінійну регресію, логістичну регресію, дерева рішень, випадкові ліси, опорні векторні машини, нейронні мережі тощо. Вибір залежить від того, чи є завдання регресією (передбачення безперервних результатів) чи класифікацією (присвоєння точок даних категоріям);

– вибір функції. На цьому етапі необхідно вирішити, які функції (незалежні змінні) включити в модель. Вибір функцій має на меті зберегти найбільш релевантні змінні, одночасно видаляючи нерелевантні або зайві. У цьому процесі можуть допомогти такі методи, як оцінювання важливості ознак або рекурсивне виключення ознак;

– навчання моделі. Для навчання моделі використовується частина даних (навчальний набір). Під час навчання модель вивчає зв’язки між вхідними характеристиками та цільовою змінною. Вибір алгоритмів і гіперпараметрів (параметрів моделі) може істотно вплинути на продуктивність моделі;

– перехресна перевірка. Методи перехресної перевірки (наприклад, *k*-кратну перехресну перевірку) використовуються, щоб оцінити можливість узагальнення моделі та запобігти переобладнанню. Перехресна перевірка передбачає поділ даних на кілька підмножин для навчання та тестування, щоб переконатися, що модель добре працює на невидимих даних;

– оцінка моделі. Під час цього етапу оцінюється ефективність моделі за допомогою відповідних показників, таких, як: середня квадратична помилка для завдань регресії або точність, прецизійність, запам’ятовування, оцінка F1 і ROC AUC для завдань класифікації. Вибір метрики оцінки залежить від конкретної проблеми;

– налаштування гіперпараметрів. Етап передбачає точне налаштування гіперпараметрів моделі для оптимізації її продуктивності. Це

може включати коригування швидкості навчання, параметрів регуляризації, глибини дерева або інших параметрів для досягнення кращих результатів;

– інтерпретабельність моделі. Для деяких застосувань, таких як відповідність нормативним вимогам або медична діагностика, інтерпретовані моделі, як-от лінійна регресія або дерева рішень, можуть бути кращими за складні моделі чорної скриньки, такі як нейронні мережі;

– ансамблеві методи. Ансамблеві методи використовуються, щоб покращити продуктивність моделі шляхом поєднання прогнозів кількох моделей. Ансамблі часто можуть досягти кращих результатів, ніж окремі моделі [3, 17];

– перевірка тестових даних. Після навчання та точного налаштування моделі оцінюється її продуктивність на окремому тестовому наборі даних, який модель раніше не бачила. Це дає оцінку того, наскільки добре модель працюватиме в реальних сценаріях.

Моделі можуть потребувати періодичного перенавчання або оновлення для врахування змін шаблонів даних.

Припущення щодо розподілу та етап розробки моделі розподілу даних є важливим мостом між аналізом даних і прийняттям рішень. Він перетворює дані на корисну інформацію та інструменти прогнозування, які можуть інформувати та спрямовувати осіб, які приймають рішення, у різних сферах, від фінансів і маркетингу до охорони здоров’я та машинобудування.

2.2.5 Прийняття рішень

Етап прийняття рішення є кульмінацією процесу розподілу даних у задачах прийняття рішень. Цей етап передбачає використання інформації, аналізу та результатів, отриманих на основі даних, для прийняття рішень.

Перед прийняттям рішень зацікавлені сторони та особи, які приймають рішення, повинні ретельно переглянути висновки та ідеї, отримані в

результаті аналізу даних і етапів перевірки гіпотез [2]. Це включає повне розуміння результатів, закономірностей, тенденцій і зв'язків, виявлених у даних. Необхідно переконатися, що запропоновані рішення узгоджуються з головними цілями та завданнями проблеми, яка вирішується.

Мають бути проведена оцінка потенційних ризиків, пов'язаних з різними варіантами рішень. Ця оцінка повинна враховувати не лише потенційні негативні наслідки, але й ймовірність і вплив цих наслідків.

Треба визначити різні доступні варіанти рішень або альтернативи. Ці параметри можуть ґрунтуватися на аналізі даних і рекомендаціях. Оцінюючи варіанти, мають бути враховані як короткострокові, так і довгострокові наслідки.

Необхідно провести аналіз витрат і вигод, щоб оцінити економічні наслідки різних варіантів рішень. Це передбачає кількісне визначення витрат, пов'язаних з кожним варіантом, і їх порівняння з очікуваними вигодами або відшкодуванням [25].

Мають бути встановлені чіткі критерії або показники, які використовуватимуться для оцінки та визначення пріоритетності варіантів рішень. Ці критерії можуть включати фінансові показники, стратегічне узгодження, здійсненність і відповідність нормам.

У деяких випадках можуть використовуватися математичні моделі або алгоритми прийняття рішень, щоб допомогти в процесі прийняття рішень. Ці моделі можуть допомогти кількісно визначити очікувані результати та допомогти у виборі найбільш сприятливого варіанту [7].

Для прийняття рішень має бути сформована команда. Краще залучати ключових зацікавлених сторін і експертів у відповідній галузі до процесу. Спільне прийняття рішень може принести різноманітні точки зору та предметні знання.

Після прийняття рішення необхідно розробити детальний план реалізації обраного рішення. Цей план має окреслювати конкретні кроки, обов'язки та часові рамки для ефективного виконання рішення.

Також необхідно розглянути потенційні виклики або перешкоди, які можуть виникнути під час реалізації рішення. Треба розробити плани на випадок надзвичайних ситуацій для завчасного вирішення цих проблем.

На етапі прийняття рішень отримані на основі даних ідеї та аналіз перетворюються на дії та стратегії. Ефективне прийняття рішень використовує дані як цінний ресурс, направляючи організації та окремих людей до оптимального вибору та результатів, мінімізуючи ризики та невизначеності.

2.2.6 Підтримка прийняття рішень і моніторинг

Після прийняття рішення проводиться періодичні перегляди для оцінки ефективності та внесення коригувань, якщо це необхідно. Процес прийняття рішень часто є повторюваним і має можливості для постійного вдосконалення.

У ситуаціях, коли своєчасне прийняття рішень має вирішальне значення, системи підтримки прийняття рішень у режимі реального часу пропонують найновіші дані та аналіз, що дозволяє швидко реагувати на зміни умов.

Системи підтримки прийняття рішень розроблені для роботи з великими обсягами даних і масштабуються в міру зростання даних. Ця масштабованість гарантує, що система залишається ефективною, оскільки вимоги до даних зростають.

Системи підтримки прийняття рішень часто включають механізми зворотного зв’язку та моніторингу продуктивності. Це дозволяє користувачам оцінювати результати рішень і за потреби коригувати стратегії.

Системи підтримки прийняття рішень підлягають постійному вдосконаленню та оптимізації. Регулярні оновлення, відгуки користувачів і технологічний прогрес сприяють їх ефективності [2].

2.3 Методика розподілу даних у задачах прийняття рішень на основі методу Гейла-Шеплі

Алгоритм Гейла-Шеплі, також відомий як алгоритм проблеми стабільного шлюбу, є широко поширеним методом для вирішення проблем зіставлення та розподілу даних в задачах, коли є два набори агентів з перевагами серед членів іншого набору [19]. Спочатку він був розроблений для вирішення проблеми підбору чоловіків і жінок у стабільних шлюбах, але його застосування поширюється на різні сфери, включаючи розподіл робочих місць, розподіл ресурсів тощо.

Перш за все необхідно вказати проблему, яку необхідно вирішити за допомогою алгоритму Гейла-Шеплі. Потрібно сформувати дві групи агентів з рівною кількістю членів у кожній групі з перевагами над членами іншої групи.

Кожен агент у першій групі повинен ранжувати агентів у другій групі на основі своїх уподобань, і навпаки. Ці переваги можуть бути представлені у вигляді списків переваг або матриць. Кожен чоловік має список жінок, а кожна жінка має список чоловіків.

Наступним кроком необхідно ініціалізувати всіх агентів як незадіяних і без будь-яких пропозицій. У контексті шлюбу всі люди спочатку неодружені.

Мета полягає в тому, щоб знайти стабільну відповідність, де жодна пара індивідів з різних груп не віддала б перевагу один одному своїм поточним партнерам.

Алгоритм Гейла-Шеплі працює наступним чином [1]:

Крок 1. Спочатку кожен чоловік робить пропозицію своїй найбільш вподобаній жінці, яка ще не відмовила йому.

Крок 2. Кожна жінка отримує кілька пропозицій і вибирає найкращу пропозицію з чоловіків, які зробили їй пропозицію, виходячи зі своїх власних уподобань.

Крок 3. Агенти другої групи, які отримують пропозиції, можуть прийняти або відхилити їх на основі власних уподобань. Якщо агент другої групи вже має пару і отримує кращу пропозицію, він розриває попереднє зіставлення та приймає нову пропозицію. Агенти, що мають пару, залишаються зі своїм партнером, доки не отримають кращу пропозицію.

Крок 4. Чоловіки, яким було відмовлено, роблять пропозицію своїй наступній найбільш вподобаній жінці, яка ще не відмовила їм.

Крок 5. Кроки 1-4 повторюються до тих пір, поки в першій групі не залишиться агентів без пари. На цьому алгоритм завершує роботу. Алгоритм гарантує, що він завершиться стабільним збігом, і кожна особа буде зіставлена з найбільш бажаним партнером, який готовий їх прийняти.

Таким чином, результатом роботи алгоритму є набір стабільних пар (збігів), де жоден агент з першої та другої групи не хотів би бути один з одним більше, ніж з їхніми поточні партнерами [1]. Ці стабільні пари представляють остаточний розподіл ресурсів або рішень.

Нижче представлено псевдокод алгоритму.

Лістинг 2.1 Псевдокод алгоритму Гейла-Шеплі:

```
function stableMatching(M, W):
  Initialize all men and women to be free
  while (there is an unmarried man m who has not proposed to every woman):
    Choose such a man m
    Let w be the highest-ranked woman in m's preference list to whom m has not yet proposed
    if (w is free):
      (m, w) become engaged
    else if (w prefers m to her current fiancé m'):
      (m, w) become engaged
      m' becomes free
    else:
      m remains free
  end while
return the list of engagements
```

Важливо відзначити, що алгоритм Гейла-Шеплі гарантує стабільне рішення, але кінцевий результат може залежати від початкових уподобань агентів. Тому вкрай важливо переконатися, що сформульовані переваги точно відображають пріоритети агентів у контексті проблеми. Крім того, цей алгоритм передбачає, що в обох групах є однакова кількість агентів.

3 ДОСЛІДЖЕННЯ МЕТОДУ ГЕЙЛА-ШЕПЛІ ДЛЯ РОЗПОДІЛУ ДАНИХ У ЗАДАЧАХ ПРИЙНЯТТЯ РІШЕНЬ

3.1 Вибір інструментальних засобів для реалізації методу Гейла-Шеплі стосовно розподілу даних у задачах прийняття рішень

Майбутній застосунок має бути сучасним та використовувати актуальні технології для подальшої підтримки.

Для реалізації методу та застосунку в якості мови програмування було обрано TypeScript. Це – мова програмування, розроблена корпорацією Microsoft, яка базується на JavaScript і розширює його. TypeScript додає статичну типізацію до JavaScript. Це дозволяє розробникам визначати типи змінних, параметрів функції та значення, що повертаються. Типізація дозволяє на ранньому етапі виявляти помилки і покращує зручність обслуговування та читання коду. Це основна мова для розробки клієнтської частини вебзастосунку [40].

Для реалізації backend частини використовується Node.js, що дозволяє виконувати TypeScript на сервері.

В якості бібліотеки для клієнтської частини обрано React. Це – бібліотека JavaScript з відкритим вихідним кодом, що призначена для розробки інтерфейсу користувача.

Для стилізації компонентів буде використовуватись MaterialUI – популярна платформа, яка реалізує рекомендації Google щодо матеріального дизайну. MaterialUI надає набір готових компонентів React, стилів та утиліт, які дозволяють розробникам створювати сучасні, адаптивні та візуально узгоджені вебзастосунки.

Аби спростити розробку серверної частини вирішено використовувати фреймворк NextJS – легкий та ефективний фреймворк для розробки серверних застосунків на JavaScript або TypeScript. Має підтримку React.

Для розробки бази даних використовується PostgreSQL – система управління реляційними базами даних. Вона дотримується стандартів SQL.

Також, для відстежування версій використовується Git. Він дозволяє відстежувати зміни в коді та співпрацювати з іншими розробниками.

Для системи керування залежностями використовується npm, що допомагає встановлювати та керувати залежностями проєкту.

В якості інструменту для розробки та налагодження обрано Visual Studio Code. Це – редактор коду з підтримкою багатьох мов програмування та розширень.

За допомагою DevTools в браузері можна відстежувати, відлагоджувати та вдосконалювати фронтенд частину застосунку.

Для тестування написаного коду обрано фреймворк Jest. Він широко використовується для тестування коду JavaScript і TypeScript, особливо в контексті програм React. Jest відомий своєю простотою, швидкістю та функціями, зручними для розробників.

3.2 Етапи програмної реалізації методу Гейла-Шеплі для розподілу даних у задачах прийняття рішень

3.2.1 Опис концепції застосунку

Суть алгоритму Гейла-Шеплі – в знаходженні стабільної відповідності між елементами двох груп. Таким чином, необхідно виділити дві групи агентів. Оскільки стоїть завдання – розробити та реалізувати програмний застосунок для розподілу волонтерів між волонтерськими проєктами, в основі якого застосовано метод Гейла-Шеплі, то групами будуть: волонтери, що шукають проєкти, та проєкти, яким потрібні волонтери.

За алгоритмом Гейла-Шеплі агенти обох груп мають розставити за пріоритетами варіанти, що їм сподобалися. Отже, з обраних проєктів волонтер має зробити список, де на першому місті має бути найбільш

цікавий для волонтера варіант. Натомість, для проєкту оцінювання кандидатів буде проводитись програмно за допомогою порівняння певних критеріїв, таких як: досвід, вміння, soft skills тощо.

Таким чином, необхідно сформувати критерії, за якими кандидати будуть порівнюватись. Серед 143 респондентів було проведено опитування з метою виявити основні навички, необхідні для ефективної участі у волонтерських проєктах. Комунікативні навички стали одностайним пріоритетом, підкреслюючи важливість спільної взаємодії в командах волонтерів. Окрім цього, особливу увагу приділяли до володіння англійською мовою як мовою міжнародного спілкування. Технічні знання, здатність до адаптації та готовність до навчання були визначені як важливі атрибути, що відображають характер волонтерських ініціатив. Крім технічних аспектів, такі навички, як командна робота, емпатія та вирішення проблем, були виділені як невід'ємні для сприяння позитивній груповій роботі. Опитування також підкреслило важливість залучення в проблематику проєкта, аби волонтерська діяльність приносила задоволення.

На основі цього опитування був сформований перелік навичок, що буде запропонований користувачам застосунку (табл. 3.1).

Таблиця 3.1 – Перелік навичок, що буде запропонований користувачам застосунку

Назва	**Опис**
1	**2**
Комунікаційні навички	Здатність ефективно передавати ідеї, активно слухати та співпрацювати з різними членами команди
Адаптивність	Бажання та здатність орієнтуватися в змінах, вивчати нові навички та пристосовуватися до нових вимог проєкту

Продовження таблиці 3.1

1	2
Вирішення проблем	Здатність аналізувати виклики та вносити інноваційні рішення для проблем проєкту
Командна робота та співпраця	Здатність гармонійно працювати в команді, розуміючи та цінуючи різноманітні точки зору
Емпатія	Глибоке розуміння потреб і перспектив інших, сприяння створенню співчутливого та сприятливого середовища в команді
Управління часом	Ефективна організація та вміння розставляти пріоритети завдань, щоб дотриматися кінцевих термінів проєкту та зобов’язань
Лідерські якості	Ініціативність, відповідальність і здатність направляти та мотивувати інших, коли це необхідно
Стійкість	Здатність вистояти перед викликами, з якими можна стикнутися на проєкті
Культурна компетентність	Обізнаність і повага до різних культур, забезпечення інклюзивності в команді волонтерів
Гнучкість	Відкритість новим ідеям, підходам і готовність адаптуватися до мінливої динаміки проєкту

Продовження таблиці 3.1

1	2
Організаційні навички	Здатність ефективно керувати завданнями, інформацією та ресурсами, щоб сприяти успіху проєкту
Відповідальності	послідовність і надійність у виконанні зобов’язань і відповідальності в рамках волонтерського проєкту
Креативність	Здатність мислити нестандартно та вносити новаторські ідеї в проєкт
Управління проєктами	Навички планування, організації та ефективного виконання завдань для забезпечення успіху проєкту
Вирішення конфліктів	Здатність конструктивно вирішувати конфлікти всередині команди
Цифрова грамотність	Вміння використовувати цифрові інструменти та платформи, пов’язані з проєктом
Володіння іноземною мовою	Залежно від контексту проєкту, знання певних мов може бути вирішальним
Міжкультурне спілкування	Здатність орієнтуватися та ефективно спілкуватися в різноманітних культурних умовах
Екологічна обізнаність	Розуміння та відданість сталим та екологічно свідомим практикам, актуальним для певних проєктів

Цей список може змінюватись, доповнюватись на основі відгуків користувачів застосунку.

Особа, що додає проєкти до застосунку, має зазначити які навички є важливими для проєкту, та вказати коефіцієнт важливості кожної. Цей коефіцієнт буде перемножуватися на оцінку навику волонтера, що була їм виставлена.

Наступним питанням є момент запуску алгоритму Гейла-Шеплі. Немає сенсу застосовувати алгоритм одразу після того, користувач завершить формування свого списку проєктів, що йому сподобалися, оскільки інші користувачі ще можуть бути в процесі формування списку. Тому необхідно запровадити певний термін для створення користувачами пріоритезованого списку. Вирішено додати дедлайни – кінцеві терміни, коли волонтери можуть додати певний проєкт до свого списку. Такими термінами є 1 та 15 числа кожного місяця, тобто двічі на місяць.

Після того, як за допомогою алгоритму зіставлені волонтери та проєкти, необхідно передбачити функціонал відгуків, аби зібрати статистику задоволеності волонтерів отриманими проєктами.

3.2.2 Програмна реалізація

Після опису логіки вебзастосунку можна приступати до його програмної реалізації. Першим кроком необхідно створити новий проєкт за допомогою виконання в терміналі команди (рис. 3.1).

```
npx create-next-app@latest --typescript
```

Рисунок 3.1 – Команда для створення нового проєкту

Дана команда пропонує налаштувати структуру проєкту та запускає завантаження необхідних пакетів (рис. 3.2).

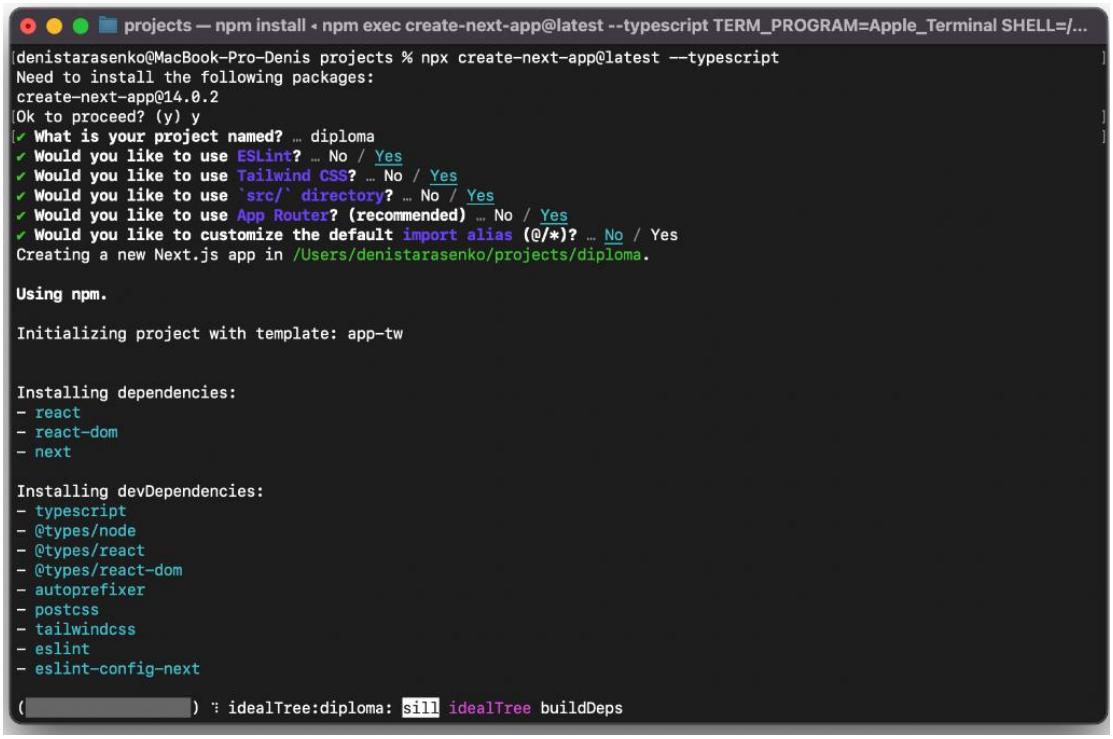

Рисунок 3.2 – Створення нового проєкту

Після завершення завантаження всіх необхідних пакетів та ініціалізації проєкту його файлова структура буде виглядати, як зображено на рисунку 3.3.

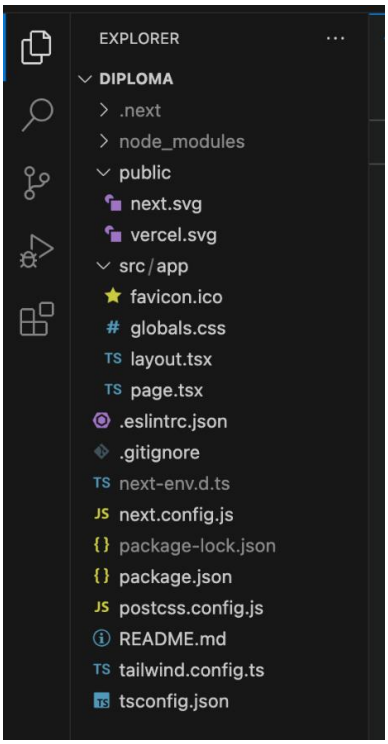

Рисунок 3.3 – Файлова структура проєкту

В папці src/app знаходяться сторінки вебзастосунку. Для зберігання власних компонентів краще створити додаткову папку, наприклад, src/components. Це дозволить повторно використовувати існуючи компоненти, що підвищить чистоту написання коду.

Коли файлова структура визначена, можна приступати до створення необхідних сторінок застосунку. Користувачі повинні мати змогу зареєструватися та увійти до свого облікового запису, якщо вони вже зареєстровані.

Для прискорення стилізації компонентів застосунку, буде використовуватись бібліотека Material UI. Таким чином, скориставшись шаблонами даної бібліотеки, були розроблені сторінки реєстрації та входу в обліковий запис користувача (рис. А.1, рис. А.2).

Всі дані застосунку будуть зберігатися в базі даних PostgreSQL. Для цього необхідно під’єднати проєкт до БД (рис. 3.4).

```
import { Pool } from 'pg';

const pool = new Pool({
  user: process.env.POSTGRES_USER,
  host: process.env.POSTRGRES_HOST,
  database: process.env.POSTGRES_DATABASE,
  password: process.env.POSTGRES_PASSWORD,
  port: process.env.POSTGRES_PORT,
});

export default pool;
```

Рисунок 3.4 – Підключення проєкту до бази даних

Для хешування паролю користувача буде використовуватись функція hash бібліотеки bcrypt (рис. 3.5).

```
import bcrypt from 'bcrypt';

export async function hashPassword(password: string): Promise<string> {
  const saltRounds = 10;
  const hashedPassword = await bcrypt.hash(password, saltRounds);
  return hashedPassword;
}
```

Рисунок 3.5 – Створення функції hashPassword для хешування паролю користувача

Після реєстрації користувачу буде запропоновано заповнити свій профіль необхідними даними, такими, як: досвід волонтерства та навички, які користувач має оцінити за 5-бальною шкалою. Результат реалізації даної сторінки зображено на рисунку А.3.

Заповнивши дані про себе, користувач отримує доступ до перегляду списку проєктів, з яких він може створити власний список проєктів, що йому сподобались. Дана сторінка зображена на рисунку А.4.

Після натискання кнопки «Подобається» (рис. 3.6) проєкт додається до списку вподобаних проєктів. Подивитися усі вподобані проєкти користувач може, перейшовши на сторінку, зображену на рисунку А.5.

Рисунок 3.6 – Кнопка «Подобається»

Користувач має призначити кожному проєкту пріоритет. Після цього, як будо визначено бізнес-логікою застосунку, в певний час запускається алгоритм Гейла-Шеплі. Код реалізації алгоритму для зіставлення волонтерів та проєктів наведено на рисунках 3.7, 3.8. За написаним алгоритмом передбачається, що на проєкті може бути декілька волонтерів, але один волонтер може бути задіяний тільки на одному проєкті в один момент часу.

```
function galeShapley(
  volunteers: Volunteer[],
  projects: Project[]
): Map<string, string[]> {
  const engagements = new Map<string, string[]>();

  while (
    volunteers.some((volunteer) => volunteer.currentProject === null) &&
    projects.some((project) => project.canAcceptMoreVolunteers())
  ) {
    for (const volunteer of volunteers) {
      if (volunteer.currentProject === null) {
        const projectId = volunteer.preferences.shift()!;
        const project = projects.find((p) => p.id === projectId)!;

        if (project.canAcceptMoreVolunteers()) {
          project.acceptedVolunteers.push(volunteer.id);
          volunteer.currentProject = projectId;

          if (!engagements.has(projectId)) {
            engagements.set(projectId, []);
          }

          engagements.get(projectId)!.push(volunteer.id);
        } else {
          const currentVolunteers = engagements.get(projectId)!;
          const currentVolunteer =
            currentVolunteers[currentVolunteers.length - 1];
```

Рисунок 3.7 – Лістинг коду реалізації алгоритму Гейла-Шеплі для зіставлення волонтерів та проєктів (початок лістингу)

```
          if (compareVolunteers(currentVolunteer, volunteer.id)) {
            project.acceptedVolunteers.pop();
            project.acceptedVolunteers.push(volunteer.id);
            volunteer.currentProject = projectId;

            currentVolunteers.pop();
            currentVolunteers.push(volunteer.id);
            volunteers.find(
              (v) => v.id === currentVolunteer
            )!.currentProject = null;
          }
        }
      }
    }
  }

  return engagements;
}
```

Рисунок 3.8 – Лістинг коду реалізації алгоритму Гейла-Шеплі для зіставлення волонтерів та проєктів (кінець лістингу)

Функціоналом передбачені наступні випадки, що можуть бути в результаті розподілу:

– деякі волонтери можуть залишитись без проєктів. Це означає, що на всіх вподобаних проєктах знайшлися кандидати краще по критеріях, зазначених проєктом;

– деякі проєкти можуть залишитись без волонтерів. Це означає, що всі волонтери були прийняті на інші проєкти через вищий пріоритет.

Для збору відгуків користувачів щодо розподілу волонтерів між проєктами розроблена форма, зображена на рисунку А.6.

На цьому розробка застосунку завершена.

3.3 Тестування розробленого застосунку та аналіз результатів

Для тестування застосунку було залучено 86 осіб, що виступили в ролі волонтерів. Для тестування було зменшено інтервал між запусками алгоритму Гейла-Шеплі, аби провести більше тестових запусків за менший проміжок часу. Завдяки формі зворотного зв’язку (рис. А.6) було зібрано відгуки користувачів.

Користувачі відзначили інтуїтивно зрозумілий інтерфейс. Були запропоновані варіанти для покращення, як, наприклад: додати можливість користувачу приймати участь у декількох проєктах за один проміжок часу. Окрім цього, користувачі запропонували змінити інтервал запуску алгоритму, оскільки користувачам необхідно чекати доволі довгий час, щоб отримати проєкт.

Після розподілу волонтерів між проєктами 12 користувачів залишилися незадоволені розподілом. Головна причина в тому, що вони залишилися без проєктів, незважаючи на те, що інші волонтери з такими ж критеріями отримали проєкт. Це зумовлено тим, що проєкт має обмеження на кількість вакантних місць. В такому випадку перевага надається тому, від кого

першого надійшла заявку в алгоритмі. Таким чином, порядок, в якому зберігаються волонтери в базі даних в застосунку, має подекуди вирішальне значення. Незважаючи на те, що кількість людей, що незадовільні результатом розподілу, не є велика, це є недоліком алгоритму Гейла-Шеплі, якому варто приділити увагу у майбутньому.

Результати тестування описано в таблиці 3.2.

Таблиця 3.2 – Результати тестування застосунку

Критерій	**Результат**
Відповідність є стабільною	Так, відповідстість відповідає поняттю стабільності, визначеному в роботі Гейла та Шеплі [1]
Справедливість розподілу	Волонтери отримали найкращий результат з огляду на вимоги проєкту до кандидата. Проте є випадки, коли волонтери лишилися без проєктів, маючи такі ж дані, що й їх конкуренти. Причиною є порядок, в якому волонтери надсилають свої заявки на проєкт. Визначено як недолік алгоритму
Правильність зупинки алгоритму	Так, алгоритм завершує роботу, коли всі волонтери мають проєкт, або коли більше немає проєктів, в яких ще є вакантні місця. Виконання однієї з умов достатньо для зупинки алгоритму
Справність при навантаженні	Так, при одночасному використанні 80 користувачами застосунок працює стабільно
Зручність використання	Так, інтерфейс застосунку є зрозумілий

3.4 Перспективи подальшої роботи

Проблема, яку вирішує алгоритм Гейла-Шеплі, досить розповсюджена та охоплює багато сфер діяльності людини. Саме тому ця тематика є актуальною.

Однією з перспективних задач є оброблення випадку, що був визначений як недолік в результаті тестування застосунку – порядок, в якому волонтери надсилають свої заявки на проєкт.

Окрім цього, на основі відгуків, отриманих під час тестування, у майбутньому може бути додано:

– можливість користувачу приймати участь у декількох проєктах за один проміжок часу. Це можливо завдяки модифікації застосованого алгоритму;

– можливість бути призначеним на проєкт без необхідності обирати проєкти до списку вподобаних та призначати їм пріоритет. Це може бути зроблено завдяки додаванню схожого функціоналу для користувачів, як для проєктів. Користувач відзначає тематики проєктів, в яких має бажання приймати участь, та призначає коефіцієнт важливості певним характеристикам проєкта. Після чого система автоматично підбирає потенційно цікаві проєкти для волонтера, ранжує їх. Отриманий таким чином список далі застосовується у алгоритмі Гейла-Шеплі;

– можливість відповідальних осіб з проєкту залишати відгуки волонтерам, оцінювати їх навички, щоб створити більш об’єктивну оцінку.

ВИСНОВКИ

У рамках кваліфікаційної роботи був розроблений застосунку для розподілу волонтерів між волонтерськими проєктами, в основі якого застосовано метод Гейла-Шеплі, що призначений для створення стабільних відповідностей.

В ході роботи було проаналізовано сучасні застосунки для розподілу даних у задачах прийняття рішень. Було класифіковано та проаналізовано існуючі методи для розподілу даних у задачах прийняття рішень, наведені приклади цих методів, недоліки та переваги. Було проаналізовано літературні джерела. Описано механізм розподілу даних у задача прийняття рішень.

Було описано особливості алгоритму Гейла-Шеплі, проаналізовано його переваги та недоліки, сфери застосування. Описано актуальність тематики.

Під час тестування було підтверджено, що результатом алгоритму є стабільні відповідності. Окрім цього, було виявлено недолік алгоритму, що полягає в важливості порядку, в якому агенти надсилають свої заявки агентам з протилежної групи.

Було сформовано перспективи подальшої роботи.

Результати дослідження апробовано у вигляді тез доповідей під час I Міжнародної науково-практичної конференції «New ways of creating scientific ideas for implementation» [41].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Gale, D., & Shapley, L. S. (1962). College Admissions and the Stability of Marriage.

2. Творошенко, І. С. (2021). Технології прийняття рішень в інформаційних системах: навч. посібник. Харків: ХНУРЕ.

3. Tvoroshenko I.S., and Gorokhovatsky V.O. (2019) Modification of the branch and bound method to determine the extremes of membership functions in fuzzy intelligent systems, Telecommunications and Radio Engineering, 78(20), pp. 1857-1868.

4. Tvoroshenko, I., & Babochkin, O. (2021). Object identification method based on image keypoint descriptors.

5. Tvoroshenko, I., & Temchur, K. (2021). Features of software application development for food recognition using deep machine learning methods.

6. Tvoroshenko, I., & Kukharchuk, V. (2021). Current state of development of applications for recognition of faces in the image and frames of video captures.

7. Daradkeh, Y. I., & Tvoroshenko, I. (2020). Technologies for making reliable decisions on a variety of effective factors using fuzzy logic. International Journal of Advanced Computer Science and Applications, 11(5).

8. Ahmad M.A., Gorokhovatskyi V., Tvoroshenko I., Vlasenko N., Mustafa S.K. (2021) The Research of Image Classification Methods Based on the Introducing Cluster Representation Param eters for the Structural Description, *International Journal of Engineering Trends and Technology*, 69(10), pp. 186-192.

9. Daradkeh Y.I., Gorokhovatskyi V., Tvoroshenko I., and Zeghid M. (2022) Cluster representation of the structural description of images for effective classification, *Computers, Materials & Continua*, 73 (3), pp. 6069–6084.

10. Gorokhovatskyi V., Tvoroshenko I., Kobylin O., and Vlasenko N. (2023) Search for visual objects by request in the form of a cluster representation for the structural image description, *Advances in Electrical and Electronic Engineering*, 21(1), pp. 19-27.

11. Гороховатський В., Передрій О., Творошенко І., Марков Т. (2023) Матриця відстаней для множини компонентів структурного опису як інструмент для створення класифікатора зображень, *Сучасні інформаційні системи*, 7(1), С. 5-13.

12. Pomazan V., Tvoroshenko I., and Gorokhovatskyi V. (2023) Handwritten character recognition models based on convolutional neural networks, *International Journal of Academic Engineering Research*, 7(9), pp. 64-72.

13. Tvoroshenko I.S., and Gorokhovatsky V.O. (2019) Intelligent classification of biophysical system states using fuzzy interval logic, *Telecommunications and Radio Engineering*, 78(14), pp. 1303-1315.

14. Tvoroshenko Irina, Ahmad M. Ayaz, Mustafa Syed Khalid, Lyashenko Vyacheslav, and Alharbi Adel R. (2020) Modification of Models Intensive Development Ontologies by Fuzzy Logic, *International Journal of Emerging Trends in Engineering Research*, 8(3), pp. 939-944.

15. Гороховатський В.О., Творошенко І.С., Чмутов Ю.В. (2022) Застосування систем ортогональних функцій для формування простору ознак у методах класифікації зображень, *Сучасні інформаційні системи*, 6(3), С. 5-12.

16. Pomazan V., Tvoroshenko I., and Gorokhovatskyi V. (2023) Development of an application for recognizing emotions using convolutional neural networks, *International Journal of Academic Information Systems Research*, 7(7), pp. 25-36.

17. Творошенко І.С., Зеленський М.О. Дослідження гібридних методів для класифікації складноструктурованих зображень. *Сучасний рух науки*: тези доповідей VIII міжнародної науково-практичної інтернет-конференції (Дніпро, 3–4 жовтня 2019 р.). Дніпро, 2019. Т. 3. С. 382-387.

18. Todo, T., Wada, R., Yahiro, K., & Yokoo, M. (2021). Lazy Gale-Shapley for Many-to-One Matching with Partial Information. In Algorithmic Decision Theory: 7th International Conference, ADT 2021, Toulouse, France, November 3–5, 2021, Proceedings 7 (pp. 390-405). Springer International Publishing.

19. Bakhrushyn, V. (2020). Зарахування вступників до закладів вищої освіти як задача багатокритеріального прийняття рішень за умов невизначеності. System technologies, 3(128), 68-79.

20. Pittel, B. (2021). One-sided version of Gale–Shapley proposal algorithm and its likely behavior under random preferences. Discrete Applied Mathematics, 292, 1-18.

21. Зубова, В. В. (2019). Теорія матчингу в економіці: прикладні аспекти застосування. Агросвіт, (13-14), 53-59.

22. Zhao, B., Xu, P., Shi, Y., Tong, Y., Zhou, Z., & Zeng, Y. (2019, July). Preference-aware task assignment in on-demand taxi dispatching: An online stable matching approach. In Proceedings of the AAAI conference on artificial intelligence (Vol. 33, No. 01, pp. 2245-2252).

23. Tvoroshenko, I., & Koriakin, I. (2021). Analysis of methods for detecting and classifying the likeness of human features.

24. Daradkeh Y.I., Gorokhovatskyi V., Tvoroshenko I., and Al-Dhaifallah M. (2022) Classification of Images Based on a System of Hierarchical Features, *Computers, Materials & Continua*, 72(1), pp. 1785-1797.

25. Tvoroshenko (2020). Information technologies for decision-making on the conditions of spatially distributed objects, in Abstracts of I International Scientific and Practical Conference. Problems and perspectives of modern science and practice, Austria. pp. 45-50.

26. Lu, Y., Xu, Y., Qu, S., Xu, Z., Ma, G., & Li, Z. (2020). Multiattribute social network matching with unknown weight and different risk preference. Journal of Intelligent & Fuzzy Systems, 38(4), 4031-4048.

27. Lee, H., Lee, H., Jung, S., & Kim, J. (2022, January). Stable marriage matching for traffic-aware space-air-ground integrated networks: A gale-shapley algorithmic approach. In 2022 International Conference on Information Networking (ICOIN) (pp. 474-477). IEEE.

28. Modi, S., M Shagari, N., & Wadata, B. (2018). Implementation of stable marriage algorithm in student project allocation. *Asian Journal of Research in Computer Science*, *1*(4), 1-9.

29. Gorokhovatskyi V., Tvoroshenko I. (2023) Identification of visual objects by the search request. *International scientific symposium «INTELLIGENT SOLUTIONS-S». Computational intelligence (results, problems and perspectives). Decision making theory: proceedings of the international symposium*, September 28, 2023, Kyiv-Uzhorod, Ukraine, pp. 25-27.

30. Kobylin O., Gorokhovatskyi V., Tvoroshenko I., and Peredrii O. (2020) The application of non-parametric statistics methods in image classifiers based on structural description components, *Telecommunications and Radio Engineering*, 79(10), pp. 855-863.

31. Tvoroshenko I., and Dziubenko M. (2020) Modern methods of analysis of the movement scheme using video detection of vehicles, *Abstracts of V International Scientific and Practical Conference «Study of modern problems of civilization» (October 19-23, 2020). Oslo, Norway*, pp. 422-428.

32. Yakovleva O., Kovač M., Ardasov V. & Yeremenko I. (2023). Study on adding functionality to the Zoom online conference system for monitoring the participant activities, *Public Administration and Regional Development*, 19(1), pp. 158-184.

33. Daradkeh Y.I., Gorokhovatskyi V., Tvoroshenko I., and Zeghid M. (2022) Tools for fast metric data search in structural methods for image classification, *IEEE Access*, 10, pp. 124738-124746.

34. Кучеренко, Є. І., Творошенко, І. С., Анопрієнко, Т. В. (2016) Моделювання та оцінювання станів складних об’єктів із застосуванням формальної логіки. Системи обробки інформації, (2), 76-82.

35. Tvoroshenko I., Gorokhovatskyi V., Kobylin O., and Tvoroshenko A. (2023) Application of deep learning methods for recognizing and classifying culinary dishes in images, *International Journal of Academic and Applied Research*, 7(9), pp. 57-70.

36. Гороховатський В.О., Творошенко І.С. (2022) Аналіз багатовимірних даних за описом у формі множини компонент: монографія. Харків: ХНУРЕ. 124 с.

37. Tvoroshenko I., and Gorokhovatskyi V. (2022) The Application of Hybrid Intelligence Systems for Dynamic Data Analysis, International Journal of Engineering and Information Systems, 6(2), pp. 40-48.

38. Tvoroshenko, I. (2019). Development of models of spatial analysis of status of interactive processes of complex systems.

39. Гороховатський В., Творошенко І., Сидоренко Д. (2021) Класифікація зображень із використанням кластерного подання. Міжнародний науковий симпозіум «Інтелектуальні рішення-С». Обчислювальний інтелект (результати, проблеми, перспективи). Теорія прийняття рішень: праці міжн. наук. симпозіуму (Вересень 29, 2021). Київ – Ужгород, С. 44-45.

40. Daradkeh Y.I., Gorokhovatskyi V., Tvoroshenko I., Gadetska S., and Al-Dhaifallah M. (2023) Statistical data analysis models for determining the relevance of structural image descriptions, *IEEE Access*, 11, pp. 126938-126949.

41. Тарасенко, Д. (2023) ВИРІШЕННЯ ЗАДАЧІ ЗНАХОЖДЕННЯ СТАБІЛЬНИХ ВІДПОВІДНОСТЕЙ ЗА ДОПОМОГОЮ АЛГОРИТМУ ГЕЙЛА-ШЕПЛІ. In *The I International Scientific and Practical Conference «New ways of creating scientific ideas for implementation», September 18-20, 2023, Varna, Bulgaria. 285 p.* (p. 277).