

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА
Факультет інформаційних технологій
Кафедра інтелектуальних технологій

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
БАКАЛАВРА

на тему:

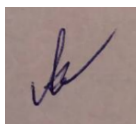
“ Система підтримки прийняття рішень при побудові функцій належності
нечіткій множині ”

Галузь знань **12 «Інформаційні технології»**

Спеціальність **122 «Комп’ютерні науки»**

Освітня програма **«Аналітика даних»**

Освітній рівень: бакалавр



Виконав: студент 4 курсу,
групи АнД-41
Айрапетов Д.А.



Керівник: Гнатієнко Г.М.
кандидат технічних наук

Випускна кваліфікаційна робота бакалавра допущена до захисту
рішенням кафедри *інтелектуальних технологій*

Протокол № 11 від 06.06.2022 р.

зав. кафедри _____ доц. Іларіонов О.Є.

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА
ШЕВЧЕНКА**

Факультет інформаційних технологій і
Кафедра інтелектуальних технологій
Спеціальність 122 «Комп'ютерні науки»

ЗАТВЕРДЖУЮ

Завідувач кафедри
інтелектуальних
технологій
Іларіонов О.Є.

“ ___ ” _____ 2021 р.

**ЗАВДАННЯ
НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ**

Айрапетову Давиду Арнольдовичу

1. Тема проекту (роботи)

Система підтримки прийняття рішень при побудові функцій належності нечіткій множині
затверджена протоколом засідання кафедри від «23 » грудня 2021 р. № 4

2. Термін здачі студентом закінченого проекту (роботи) 31 травня 2022 року

3. Вихідні дані до проекту (роботи)

Система підтримки прийняття рішень при побудові функцій належності нечіткій множині
створюється для побудови функцій належності нечіткій множині при апостеріорному
визначенні компетентності експертів в умовах невизначеності

4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)

Вибір та аналіз предметної області, постановка задачі визначення функцій належності
нечіткій множині для апостеріорного визначення компетентності експертів в умовах
невизначеності, огляд методів побудови функцій належності, проектування архітектури
системи, розробка системи, вибір тестових задач, візуалізація результатів, тестування роботи
системи

5. Перелік презентаційного матеріалу (з точним зазначенням обов'язкових презентацій)

Огляд джерел інформації щодо побудови функцій належності, ілюстрація ситуації прийняття рішення, архітектура системи, програмна реалізація системи, презентація Power Point

6. Консультанти з випускної кваліфікаційної роботи із зазначенням розділів випускної кваліфікаційної роботи, що їх стосуються

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Вибір та аналіз предметної області		
2	Постановка задачі визначення функцій належності нечіткій множині для апостеріорного визначення компетентності експертів в умовах невизначеності		
2	Огляд методів визначення функцій належності нечіткій множині		
3	Проектування архітектури системи		
3	Програмна реалізація системи		
3	Вибір тестових задач, тестування роботи системи		

7. Дата видачі завдання 25 січня 2022 року

Керівник _____ /

(підпис)

(ПІБ)

Завдання прийняв до виконання _____ /

(підпис)

(ПІБ)

КАЛЕНДАРНИЙ ПЛАН

По р. №	Назва етапів випускної кваліфікаційної роботи	Термін виконання етапів випускної кваліфікаційної роботи	Примітка
1.	Вибір та аналіз предметної області	25.01-01.02	
2.	Постановка задачі визначення функцій належності нечіткій множині для апостеріорного визначення компетентності експертів в умовах невизначеності	02.02-04.02	
3.	Огляд методів визначення функцій належності нечіткій множині	05.02-14.02	
4.	Проектування архітектури системи	15.02-07.03	
5.	Програмна реалізація системи	08.03-25.04	
6.	Вибір тестових задач, тестування роботи системи	26.04-07.05	

Студент-дипломник _____ /

/

(підпис)

(ПІБ)

Керівник випускної кваліфікаційної роботи _____ /

/

(підпис)

(ПІБ)

Анотація

Айрапетов Давид Арнольдович виконав випускню кваліфікаційну роботу на тему «Система підтримки прийняття рішень при побудові функцій належності нечіткій множині» за спеціальністю 122 – «Комп'ютерні науки».

Розглянуто проблему визначення відносної компетентності експертів у експертно-оціночних завданнях. Представлено методи визначення компетентності та продемонстровано використання об'єктивних методів для визначення цих коефіцієнтів апостеріорно. Представлено задачу визначення результативного ранжування об'єкта та метод розрахунку її розв'язання. На основі отриманого аналізом рішення задачі групового ранжирування запропоновано метод визначення коефіцієнта відносної компетентності експертів, який існує у вигляді функції, що належить до нечіткої множини. Розглянуто приклад, який ілюструє дію та особливості застосування описаного методу.

Ключові слова: експертне оцінювання, результуюче ранжування, компетентність експертів, прийняття рішень, функція належності нечіткій множині.

Summary

Airapetov David Arnoldovich completed the final qualifying work on "DSS in the construction of functions of belonging to the fuzzy set" in the specialty 122 - "Computer Science".

The problem of determining the relative competence of experts in expert evaluation tasks is considered. Methods for determining competence are demonstrated and the use of objective methods for determining these coefficients a posteriori is demonstrated. The problem of determining the effective ranking of the object and the method of calculating its solution are presented. Based on the analysis of the solution of the group ranking problem, a method for determining the coefficient of relative competence of experts is proposed, which exists in the form of a function belonging to a fuzzy set. An example illustrating the action and features of the application of the described method is considered.

Key words: expert evaluation, resulting ranking, competence of experts, decision making, function of belonging to fuzzy set.

Зміст

1. Вступ.....	8
2. Розділ 1 Аналітичний огляд та постановка задачі.....	9
2.1 Аналітичний огляд.....	9
2.2 Постановка задачі.....	11
2.2.1 Визначення результуючого ранжування при груповому оцінюванні.....	11
2.2.2 Задача визначення компетентності експертів на основі аналізу заданих індивідуальних експертних ранжувань об'єктів.....	15
2.2.3 Спосіб визначення функції належності нечіткій множині шляхом аналізу частотності значень.....	17
2.2.4 Формування вимог до СППР при побудові функцій належності нечіткій множині.....	18
3. Розділ 2. Проектування Система підтримки прийняття рішень при побудові функцій належності нечіткій множині.....	20
3.1 Огляд фреймворку для розробки.....	20
3.2 Функціональний аналіз.....	24
3.3 Аналіз процесу прийняття рішень.....	24
2.3.1 IDEF0 для СППР при побудові ФН нечіткій множині.....	24
2.3.1 Декомпозиція IDEF0.....	25
3.4 Розробка архітектури Система підтримки прийняття рішень при побудові функцій належності нечіткій множині.....	27
3.4.1 Розробка інтерфейсу користувача.....	27
3.4.2 Розробка коду для знаходження медіан за обраними методами.....	27
3.4.3 Перетворення отриманих медіан у таблиці з ідеалізованими та нормованими значеннями вагових коефіцієнтів відносної компетентності експертів.....	28

3.4.4	Перетворення таблиць з ідеалізованими та нормованими значеннями вагових коефіцієнтів відносної компетентності експертів в таблиці з	урахуванням частотності.....	29
3.4.5	Перетворення отриманих даних в ФН нечіткій множині.....		29
3.4.6	Зображення отриманих нечітких значень у вигляді графіків... ФН.....		29
2.5	Перспективи подальших досліджень.....		30
4.	Розділ 3 Розробка програмного забезпечення Система підтримки прийняття рішень при побудові функцій належності нечіткій множині.....		32
4.1	Структура програмного забезпечення.....		32
4.2	Опис структури інтерфейсу користувача.....		33
4.3	Опис результатів роботи застосунку.....		33
5.	Висновок.....		48
6.	Джерела.....		49
7.	Додатки.....		51

Вступ

Визначення коефіцієнта компетентності експертів є важливою частиною вирішення проблеми експертної оцінки. Від точності визначення відносної компетентності експертів залежить результат вирішення проблеми, надійність рішення, достовірність результатів. Уміння розглядати експертів є запорукою якості рішення і суттєво впливає на результати рішення в задачах експертної оцінки. Тому вивчення експертної компетентності та її розумне врахування є важливою сферою дослідження, яка може суттєво сприяти підвищенню адекватності та валідності аналізу результатів експертної оцінки.

Розділ 1 Аналітичний огляд та постановка задачі

1.1 Аналітичний огляд

Визначення компетентності експертів є важливим елементом експертного оцінювання. Цей чинник суттєво впливає на результати прийняття рішення і потребує поглибленого вивчення, дослідження та розвитку. На сьогодні існує кілька підходів до визначення коефіцієнтів відносної компетентності експертів, але кожен з напрямків не є досконалим [5]:

– *документальний* нерідко викликає недовіру до способів формалізації та врахування об'єктивних даних про експертів, а також визначається характерною для сучасності тенденцією до девальвації офіційних документів та некритичністю деяких інституційних рішень;

– *самооцінка*, при проведенні якої нерідко отримується інформація про рівень самовпевненості експерта, а не про його реальну компетентність;

– *взаємооцінка*, з допомогою якої у деяких задачах можна виявити конфронтацію в експертній комісії та наявності коаліцій серед її членів, які інколи спотворюють справжню індивідуальну компетентність експертів, яка дійсно має впливати на результати;

– *комбіновані методи* інколи замість синергетичного ефекту призводять до накопичення сукупної помилки;

– *об'єктивні підходи*, серед яких є зручні для застосування і достатньою мірою обґрунтовані, є найбільш надійними для обчислення адекватних показників відносної компетентності експертів.

Серед об'єктивних підходів розрізняють апріорні та апостеріорні методи обчислення компетентності. Зупинимось на методах *апостеріорного визначення* компетентності експертів, розроблених авторами. До об'єктивних підходів апостеріорного визначення коефіцієнтів компетентності експертів відносяться:

- дослідження результативності участі експерта у попередніх експертизах;

- обчислення компетентності за результатами контрольної експертизи;

- дослідження результатів участі експерта у конкретній експертизі.

Останній підхід, який полягає у дослідженні участі членів експертної групи у конкретній експертизі, може, у свою чергу, розглядатися як:

- непрямий аналіз показників без попереднього агрегування результатів;

- співвідношення початкових індивідуальних експертних даних та обчислених інтегральних показників.

В задачах визначення групового ранжування об'єктів складовими підходу, при якому досліджуються співвідношення індивідуальних експертних даних та обчислених інтегральних показників, є:

- обчислення відстаней до матриць попарних порівнянь (МПП) на основі знайденого результуючого ранжування (в різних метриках та за різними критеріями);

- апостеріорне визначення на основі результуючого ранжування об'єктів за заданими індивідуальними експертними ранжуваннями:

- = аналіз кількості інверсій у експертному ранжуванні по відношенню до результуючого ранжування;

- = визначення кількості несуперечливо проранжованих експертом об'єктів;

- = обчислення суми модулів різниць між рангами об'єктів у результуючому ранжуванні та заданих експертами ранжуваннях;

- визначення індивідуальних відносних оцінок для колективно заданих відносних оцінок:

- = обчислення суми інтервалів відносних оцінок;

- = визначення максимального інтервалу серед відносних оцінок;

- = обчислення «об'єму» інтервалів визначених відносних оцінок;

- = обчислення «площі» інтервалів відносних оцінок;

= визначення рівня відхилення МПП від заданого «ідеального» вектора відносних оцінок;

– відстань до групового вектора вагових коефіцієнтів при заданні експертами МПП у кардинальних шкалах;

– відносна відстань від експертних ранжувань до результуючого ранжування.

Моделювання та дослідження останніх двох підходів розглядається у цій роботі. Для таких підходів характерною є ситуація невизначеності. Вона породжується наявністю значної кількості розроблених та обґрунтованих різними авторами методів апостеріорного обчислення коефіцієнтів відносної компетентності експертів[6]. У той же час для конкретних практичних ситуацій неможливо достовірно та однозначно визначити – який з розроблених методів є найбільш прийнятним. Тому авторами пропонується на попередніх етапах аналізу застосовувати усі відомі методи визначення відносної компетентності експертів і лише на останніх етапах аналізу здійснити агрегування одержаної різними методами інформації.

1.2. Постановка задачі

1.2.1. Визначення результуючого ранжування при груповому оцінюванні

Методи обробки експертної інформації поділяються на три основні групи [1]:

- статистичні методи;
- методи шкалювання;
- алгебраїчні методи.

Суть алгебраїчних методів полягає в тому, що на множині допустимих оцінок задається відстань і результуюча оцінка визначається як така, відстань якої до оцінок експертів за певним вибраним критерієм є мінімальною. На

основі алгебраїчного підходу будемо визначати коефіцієнти відносної компетентності експертів.

Нехай k експертів з індексами $l \in L = \{1, \dots, k\}$, $k > 1$, задають свої переваги на множині n об'єктів A у вигляді ранжувань об'єктів $R^l, l \in L$.

Для визначення відстаней між ранжуваннями об'єктів використовують:

– метрику Кука неспівпадання рангів об'єктів у індивідуальних ранжуваннях

$$d(R^j, R^l) = \sum_{i \in I} |r_i^j - r_i^l|, \quad (1)$$

де r_i^l – ранг i -го об'єкта у ранжуванні l -го експерта $R^l, l \in L, 1 \leq r_i^l \leq n$,

– метрику Хемінга

$$d(B^j, B^l) = 0,5 \sum_{i \in I} \sum_{s \in I} |b_{is}^j - b_{is}^l|, \quad (2)$$

де $B^l = (b_{is}^l)$, $l \in L, i, s \in I$, – МПП, що відповідають ранжуванням $R^l, l \in L$.

Інколи для задач визначення результуючого ранжування використовується евклідова метрика.[7]

Поширеною є така задача колективного експертного оцінювання. Знайти результуюче (колективне, групове, компромісне, інтегральне, агреговане, узгоджене, згорнуте, синтезоване, узагальнене, глобальне тощо) ранжування об'єктів, яке за деяким критерієм є «найближчим» до усіх експертних ранжувань. Найбільш обґрунтованим методом знаходження результуючого ранжування об'єктів вважається обчислення медіани заданих ранжувань.

Найпоширенішим методом знаходження результуючого ранжування об'єктів вважається обчислення медіани заданих ранжувань.

Позначимо множину усіх можливих ранжувань n об'єктів через Ω^R , а множину МПП, які відповідають усім можливим ранжуванням n об'єктів – через Ω^B . Множину заданих експертами ранжувань будемо позначати через R^A

, а множину відповідних їм МПП – через R^B .

Для випадку, який розглядається у цій роботі, потужність множин R^A та R^B однакова: $|R^A| = |R^B| = n$, $R^l \in R^A$, $B^l \in R^B, l \in L$. Зрозуміло, що $R^A \subset \Omega^R$, $R^B \subset \Omega^B$.

У загальному випадку потужність множини $|\Omega^B| = 2^{n(n-1)/2}$. Але для методу, який описується у цій роботі, будемо вважати, що $|\Omega^R| = |\Omega^B| = n!$, оскільки нас не цікавлять нетранзитивні елементи простору розв'язків Ω^B .

Для метрики Кука (неспівпаданя рангів об'єктів) (1) при використанні утилітарного критерію обчислюється медіана Кука-Сейфорда [1]:

$$R^{CS} \in \Omega^{CS} = \mathop{Arg \min}_{R \in \Omega^R} \sum_{l \in L} d(R, R^l) \quad (3)$$

Модифікована медіана Кука-Сейфорда:

$$R^{MCS} \in \Omega^{MCS} = \mathop{Arg \min}_{R \in R^A} \sum_{l \in L} d(R, R^l) \quad (4)$$

При використанні егалітарного критерію обчислюється ГВ-медіана (компроміс) [1]:

$$R^{GB} \in \Omega^{GB} = \mathop{Arg \min}_{R \in \Omega^R} \max_{l \in L} d(R, R^l) \quad (5)$$

Модифікована ГВ-медіана:

$$R^{MGB} \in \Omega^{MGB} = \mathop{Arg \min}_{R \in R^A} \max_{l \in L} d(R, R^l) \quad (6)$$

Для метрики Хемінга (2) при використанні утилітарного критерію обчислюється медіана Кемені-Снелла:

$$R^{KC} \in \Omega^{KC} = \mathop{Arg \min}_{B \in \Omega^B} \sum_{l \in L} d(B, B^l) \quad (7)$$

Модифікована медіана [2] Кемені-Снелла:

$$R^{MKC} \in \Omega^{MKC} = \mathop{Arg \min}_{B \in R^B} \sum_{l \in L} d(B, B^l) \quad (8)$$

При використанні егалітарного критерію обчислюється ВГ-медіана

(компроміс) [3]:

$$R^{BG} \in \Omega^{BG} = \mathop{Arg \min}_{B \in \Omega^B} \max_{l \in L} d(B, B^l). \quad (9)$$

Модифікована ВГ-медіана:

$$R^{MBG} \in \Omega^{MBG} = \mathop{Arg \min}_{B \in R^B} \max_{l \in L} d(B, B^l). \quad (10)$$

Можуть враховуватися коефіцієнти компетентності експертів: ρ_1, \dots, ρ_k .

Методи визначення медіан виду (3), (5), (7), (9) та їх особливості розглядаються у монографії [3]. Модифіковані медіани (4), (6), (8), (10) пропонуються у деяких роботах [2], і їх застосування у багатьох практичних задачах є недоцільним, а інколи – необґрунтованим і неправомірним.

Модифікована медіана суттєво обмежує простір вибору, тому при застосуванні таких критеріїв, як правило, знаходимо неефективні розв'язки: вони домінуються, зокрема власне медіанами виду (4), (6), (8), (10). Крім того, метод, описаний у цій роботі, не може бути застосований для модифікованих медіан, а пошук модифікованих медіан є недоцільним та необґрунтованим у більшості випадків.

Критеріальні функції, які застосовуються для визначення медіан (3)-(10), пов'язані з відстанями від заданих експертами ранжувань до обчислених розв'язків задачі. Тому ознакою ефективності одержаного розв'язку задачі можуть слугувати мінімальні значення критеріїв (5), (6), (9), (10) для відповідних постановок задачі. Очевидно, що розв'язки, які не доставляють мінімумів зазначеним критеріям, є неефективними – вони домінуються розв'язками, які доставляють мінімум критеріям виду (5), (6), (9), (10).

Метрики та критерії для визначення медіани заданих ранжувань об'єктів можуть мати різний ступінь популярності серед дослідників, користуватися неоднаковим рівнем довіри, бути різною мірою обґрунтованими. Але усі ці інструменти належать до усталених способів визначення результуючого

ранжування і немає сенсу нехтувати будь-яким із відомих підходів, які протягом десятиліть зарекомендували себе.

Класичні методи теорії вибору (Кондорсе, Борда, Сімпсона, Копленда, Нансона, альтернативних голосів, відносної більшості тощо) також можуть бути обмежено застосовані при визначенні результуючого ранжування. Таке застосування описано, наприклад, в монографії [3]. Дослідження способів застосування класичних методів теорії вибору для визначення вагових коефіцієнтів слід дослідити при подальшому розвитку описаного в цій роботі методу.

На основі обчислення та аналізу визначених вище медіан (3), (5), (7), (9) пропонується обчислювати коефіцієнти відносної компетентності експертів у вигляді функції належності нечіткій множині (ФН).

1.2.2 Задача визначення компетентності експертів на основі аналізу заданих індивідуальних експертних ранжувань об'єктів

На сьогодні в напрямку експертного оцінювання розроблено та обґрунтовано значну кількість методів визначення колективних експертних оцінок та обчислення відносної компетентності експертів. Усі вони мають право на існування і можуть бути використані для прийняття рішень. Для визначення компетентності експертів при розв'язанні конкретної задачі експертного оцінювання авторами пропонується скористатися аналізом існуючих методів апостеріорного обчислення коефіцієнтів відносної компетентності експертів. Розроблений різноманітний апарат визначення групових оцінок виду (3)-(8) будемо використовувати та «згортати» в одну ФН.

Розглянуто одну з найважливіших задач експертного оцінювання, якою є визначення компетентності експертів. В основу запропонованих методів визначення зазначених коефіцієнтів покладено аксіому незміщеності: «Судження більшості компетентні» та її наслідок про те, що «найбільш

компетентним є той експерт, судження якого у більшості випадків співпали з висновками більшості експертів».

Особливістю неузгоджених МПП та множини ранжувань об'єктів є велика кількість ефективних розв'язків у задачі визначення групового ранжування об'єктів. Зокрема, кожна з обчислених медіан (3)-(8) може бути неєдиною.

Будемо виходити з того, що компетентність експертів визначається на основі заданих ними ранжувань об'єктів.

На кількість розв'язків задачі визначення результуючого ранжування впливають такі фактори:

- множина метрик, що застосовуються $S^1 = \{d^1, d^2, \dots\}$;
- множина критеріїв визначення близькості між ранжуваннями $S^2 = \{\delta^1, \delta^2, \dots\}$;
- неєдиний розв'язок задачі визначення медіани виду (3)-(5);
- застосування для визначення результуючого ранжування множини класичних правил вибору: Кондорсе, Борда, Сімпсона, Копленда, Нансона, альтернативних голосів, відносної більшості тощо [1], множину яких позначимо через S^3 .

Таким чином, кількість розв'язків задачі визначення колективного

ранжування об'єктів буде дорівнювати $s = \sum_{i \Leftrightarrow j} s_i^1 s_j^2 + s^3$, де символ " $i \Leftrightarrow j$ " означає «індекси метрик $i \in S^1$, для яких існують індекси критеріїв близькості $j \in S^2$ », $s^t = |S^t|$, $t = 1, 2, 3$, а через $|\cdot|$ – кількість елементів множини.

В результаті застосування усіх способів обчислення відносних коефіцієнтів компетентності експертів одержуємо набір ранжувань об'єктів, які є медіанами заданих експертних ранжувань[8]:

$$R^i \in \Omega^K = \Omega^{CS} \cup \Omega^{FB} \cup \Omega^{KC} \cup \Omega^{BF} \cup \Omega^3, \quad i = 1, \dots, s.$$

При апостеріорному визначенні відносних коефіцієнтів компетентності у задачах ранжування мірою компетентності слугують відстані від експертних ранжувань до обчислених медіан. Тобто нормовані вагові коефіцієнти відносної компетентності експертів на основі знайдених s результуючих ранжувань, які складають множину Ω^K , будемо обчислювати за формулою:

$$\gamma_{ij} = (1/d(R^i, R^j)) / \sum_{t=1}^s (1/d(R^t, R^j)), \quad R^i \in \Omega^K, \quad i = 1, \dots, s, \quad j \in I. \quad (11)$$

Ідеалізовані вагові коефіцієнти [4] відносної компетентності критеріїв будемо обчислювати за формулами:

$$\gamma_{ij}^{id} = \gamma_{ij} / \max_{t \in I} \gamma_{it}, \quad i = 1, \dots, s, \quad j \in I. \quad (12)$$

Оскільки варіантів визначення результуючого ранжування можуть бути десятки, коефіцієнти компетентності можуть бути обчислені у вигляді ФН.

Необхідність у визначенні коефіцієнтів відносної компетентності експертів у вигляді ФН обумовлена різними причинами. Не зважаючи на те, що застосування різних метрик та критеріїв для визначення результуючого ранжування обґрунтовані різною мірою, але усі вони успішно застосовуються у різних прикладних ситуаціях. Відмова від використання деяких інструментів тягне за собою ризик втратити інформацію щодо структури заданих експертами відношень переваги; у нашому випадку – лінійного упорядкування об'єктів. Щоб цього не відбулося, усі можливі розв'язки, одержані різними підходами, збираються в один масив для їх подальшого аналізу та підготовки висновків.

1.2.3 Спосіб визначення функції належності нечіткій множині шляхом аналізу частотності значень

Для застосування методу визначення ФН за частотністю значень слід визначитися з універсальною множиною.[10] Для вимог цієї роботи, універсальною множиною для визначення ФН є значення з інтервалу (0,1), а

саме – результати визначення відповідних нормованих коефіцієнтів відносної компетентності експертів на основі обчислення величин (11), обернених до відстаней між ранжуваннями. Для ідеалізованих значень вагових коефіцієнтів виду (12) універсальною множиною є ідеалізовані значення коефіцієнтів, які також вибираються з інтервалу $(0,1)$.

Нехай необхідно проаналізувати результати s числових оцінок чи результатів вимірювання чи оцінювання деякої величини та визначити ФН для вимірних або оцінених значень. Як правило, групування даних здійснюється за 10–20 інтервалами. При цьому кількість значень, які потрапляють до кожного інтервалу, не перевищує 15–20 % від їх загальної кількості. Цього виявляється достатньо для виявлення усіх властивостей величини та надійного обчислення за груповою частотністю основних характеристик нечіткої множини. У випадках, коли кількість інтервалів перевищує 20, ФН може виявитися полімодальною, а відображена нею інформація не буде наочним представленням нечіткої множини. Групування даних за надто великими інтервалами може призвести до втрати уявлення експерта про поведінку ФН, а також до грубих помилок при її застосуванні.

Очевидно, що шляхом аналізу частотності значень кількість значень оцінюваної величини, які потрапляють у кожний вибраний інтервал, свідчить про міру належності цієї величини до кожного інтервалу. У цьому випадку фактично здійснюється лише класифікація точок за їх значеннями: потрапляє чи не потрапляє точка до вибраного інтервалу. Кількість точок, що потрапили до інтервалу, свідчить про міру оптимальності чи квазіоптимальності цього інтервалу. Це обґрунтовує застосування алгоритмів частотності при визначенні ФН.

Відомо, що для номінальних ознак, тобто вимірних у шкалі найменувань, в якості середнього застосовується мода. Для даних, вимірних у порядковій шкалі, допустимим середнім є медіана. Коли досліджується інформація, виміряна в шкалі інтервалів, можливо використовувати лише

середнє арифметичне. А для аналізу даних, заданих у шкалі відношень, застосовуються ступеневі середні та середнє геометричне.

При побудові ФН за частотністю значень ми фактично використовуємо результати аналізу моди оцінених значень досліджуваної величини.

1.2.4 Формування вимог до СППР при побудові функцій належності нечіткій множині

До СППР при побудові функцій належності нечіткій множині висувуються наступні функціональні вимоги:

1. Можливість введення вхідних даних.
2. Розрахунок медіан для заданої множини експертних ранжувань.
3. Табличне зображення результатів обчислення медіан.
4. Побудова таблиць нормованих та ідеалізованих значень вагових коефіцієнтів компетентності експертів з урахуванням частотності значень.
5. Розрахунок таблиць частотності нормованих та ідеалізованих значень вагових коефіцієнтів відносної компетентності експертів у вибраних інтервалах.
6. Формування сингтонів частотності нормованих та ідеалізованих значень вагових коефіцієнтів.
7. Здійснення нормалізації функцій належності і представлення нечітких значень відносних коефіцієнтів компетентності експертів у вигляді нормальних функцій належності.

Розділ 2. Проектування Система підтримки прийняття рішень при побудові функцій належності нечіткій множині

2.1 Огляд фреймворку для розробки

JavaScript — це багатопарадигмальна мова програмування. Підтримуються об'єктно-орієнтований, імперативний та функціональний стилі. JavaScript часто використовується як вбудована мова для програмного доступу до програмних об'єктів. Він широко використовується в браузерах як мова сценаріїв, щоб зробити веб-сторінки інтерактивними. JavaScript є об'єктно-орієнтованою мовою, але прототипування, що використовується в цій мові, спричиняє відмінності в обробці об'єктів порівняно з традиційними мовами, орієнтованими на клас. Крім того, JavaScript має багато властивостей, притаманних функціональним мовам - функції як об'єкти першого класу, об'єкти як списки, турботи, анонімні функції, блокування - які надають мові додаткову гнучкість.

CSS — це формальна мова, яка використовується для опису зовнішнього вигляду документів (веб-сторінок), написаних мовами розмітки. Творці веб-сторінок використовують CSS для визначення кольорів, шрифтів, стилів, окремих блоків та інших аспектів відтворення цих веб-сторінок. Основною метою розробки CSS є інкапсуляція та відокремлення опису логічної структури веб-сторінки від опису зовнішнього вигляду веб-сторінки. Такий поділ може підвищити доступність документів, забезпечити більшу гнучкість і керованість їх подання, а також зменшити складність і повторюваність структурованого вмісту. Крім того, CSS дозволяє відображати один і той же документ у різних стилях або методах виведення, наприклад, відображення екрана, друкування, читання мовлення.

React - це JavaScript-бібліотека для розробки інтерфейсу користувача. React — це декларативна, ефективна та гнучка бібліотека JavaScript для створення інтерфейсів користувача. Він дозволяє збирати складні інтерфейси

користувача з невеликих ізольованих фрагментів коду, які називаються «компонентами».

```
<!DOCTYPE html>
<html>
  <head>
    <title>Parcel Sandbox</title>
    <meta charset="UTF-8" />
  </head>

  <body>
    <div class="root">
      <h1 id="counter-value"></h1>
      <button id="increment-btn">+1</button>
    </div>

    <script>
      const counterValueElement = document.getElementById("counter-value");
      const incrementBtn = document.getElementById("increment-btn");
      let counterValue = 0;

      function increment() {
        counterValue += 1;
        counterValueElement.innerText = counterValue;
      }

      counterValueElement.innerText = counterValue;
      incrementBtn.addEventListener("click", increment);
    </script>
  </body>
</html>
```

Рисунок 2.1 Приклад коду

Весь код знаходиться у функції App. У React це та інші подібні функції називаються компонентами. Компоненти – це фрагменти інтерфейсу, які містять розмітку і, якщо необхідно, пов’язану логіку. Усі програми React побудовані на компонентах. Сам компонентний підхід був задовго до React, але він поєднувався з декларативним. характерна реакція React не є універсальним інструментом для будь-якого проекту. Щоб зрозуміти, чи може React вирішити вашу проблему, важливо зрозуміти його сильні та слабкі сторони. Ця бібліотека може полегшити життя розробникам: Ви можете використовувати його для

створення простих у обслуговуванні інтерфейсів для окремих компонентів. Він додає зручний шар абстракції без необхідності безпосередньо працювати з DOM.

React більше не є новою бібліотекою, вона підтримується великою компанією Facebook і величезною спільнотою розробників. Таким чином, він добре перевірений, регулярно обслуговується та постійно оновлюється, а перехід до нових версій відбувається максимально плавно. Завдяки спільноті React має добре задокументований і багатий досвід зі статтями, курсами та конференціями. Це не тільки полегшує початківцям знайомство з бібліотекою, а й дає змогу знайти відповіді на різноманітні запитання. На Github ви можете знайти готові компоненти React майже на будь-який випадок. Якщо у вас їх немає, але вам потрібні окремі бібліотеки, ви можете пошукати інтеграції або зробити це самостійно. З часом спільнота React розробила певні підходи та протоколи для організації проекту, коду та вирішення поширених проблем. Для розробників це означає, що вам потрібно витратити менше часу на обговорення деяких протоколів – ви можете використовувати стандартні та вже встановлені. Хоча проект може мати процес збирання, який використовує веб-пакет, посилку, згортання чи інші пакети, пам'ятайте, що це необов'язкове для React. З цією бібліотекою ви пишете на чистому JS і вам не потрібно вивчати діалекти HTML, CSS або JS. Звичайно, JSX майже завжди використовується з React, але він також необов'язковий. React — це проект з відкритим кодом. Це робить його безпечним для використання навіть у комерційних додатках. Однак є деякі важливі особливості, які слід враховувати при використанні React: React збільшує розмір програми, яку користувачі повинні завантажити (пакети React і React-dom мають близько 40 КБ). Завантажений код має виконуватися в браузері: це означає, що програма працює повільніше. Детальніше про це можна прочитати в матеріалі JavaScript у 2019 році командою V8 (переклад: Ціна JavaScript у 2019 році).

На додаток до віртуальної DOM є її накладні витрати: по-перше, час виконання (порівняння віртуальних дерев не є миттєвим); по-друге, з пам'яті, оскільки віртуальні дерева потрібно зберігати десь, а не в одній копії. Зі збільшенням кількості елементів на сторінці збільшуються ці витрати, що може стати справжньою проблемою на мобільних пристроях. Тому, вивчаючи React, важливо звернути увагу на способи оптимізації візуалізації програм — для цього в бібліотеці є необхідні інструменти.

Зрозуміло, ці недоліки не є причиною повністю відмовлятися від React і подібних бібліотек при створенні проекту. Але для того, щоб свідомо використовувати цей інструмент і розуміти його обмеження, їх потрібно пам'ятати.

Чому я обрав React:

По-перше, React – це популярна бібліотека, яка залишиться актуальною ще довго. Це означає, що ви завжди можете знайти проект, написаний на React, і працювати над ним.

По-друге, це чудовий новий світ з величезною спільнотою. React допомагає поглянути на розробку інтерфейсу зовсім по-іншому через призму декларативного програмування. Це корисно для загального розвитку та розширення світогляду, а отримані знання спростять дослідження інших подібних бібліотек і технологій (Vue.js, Svelte і навіть SwiftUI). Крім того, багато протоколів, шаблонів і методів на основі спільноти допомагають писати найкращий і найбільш підтримуваний код. Але також важливо розуміти, що React — це бібліотека для мови JS. І тому потрібно добре володіти JS, HTML і CSS.

2.2 Функціональний аналіз

Карта процесів (рис. 2.2)

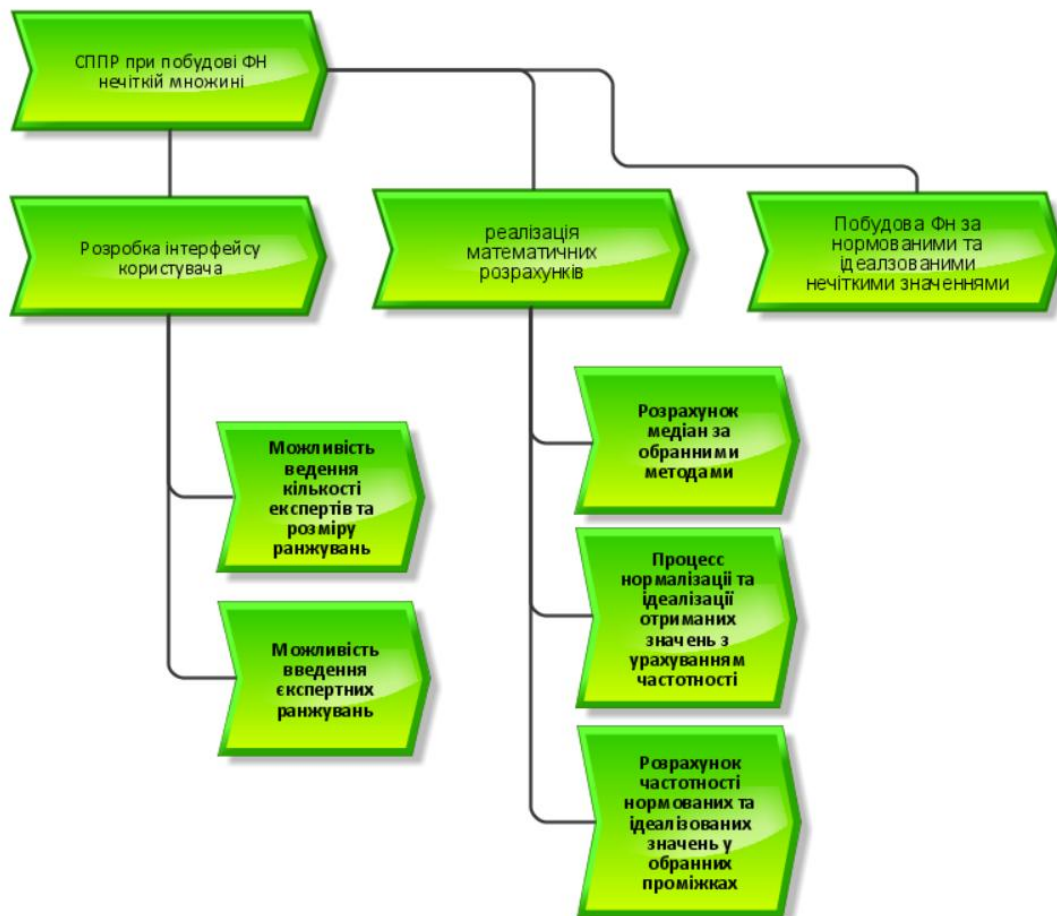


Рисунок 2.2 Карта процесів для системи підтримки прийняття рішень при побудові ФН нечіткій множині

2.3 Аналіз процесу прийняття рішень

2.3.1 IDEF0 для СППР при побудові ФН нечіткій множині (рис. 2.3)

IDEF0 – Підхід до функціонального моделювання. За допомогою мови візуальної графіки IDEF0 досліджувана система постає перед розробниками та аналітиками у вигляді набору взаємопов'язаних функцій (функціональних блоків – у термінах IDEF0). Загалом, моделювання IDEF0 це перший крок у вивченні будь-якої системи;



Рисунок 2.3 IDEF0 діаграма

На вхід системи поступають:

- кількість експертів; розмір експертних ранжувань; експертні ранжування.

Елементи керування:

- Научна література – математичні методи реалізації поставленої задачі;
- Вказівки керівника – обрання оптимальних методів та інші корегування.

Механізми:

- VS Code- середовище для написання коду СППР. Механізмом виконання є код який в результаті надає ФН нечіткій множині які зображають компетентність експертів.

2.3.1 Декомпозиція IDEF0 (рис 2.4; рис 2.5)

Принцип декомпозиції застосовується, коли складний процес розкладається на його функції. При цьому рівень деталізації процесу безпосередньо визначається розробником моделі. Декомпозиція дозволяє

представити модель системи покроково та структурно у вигляді ієрархії окремих діаграм, що робить її переважаною та легкою для сприйняття.

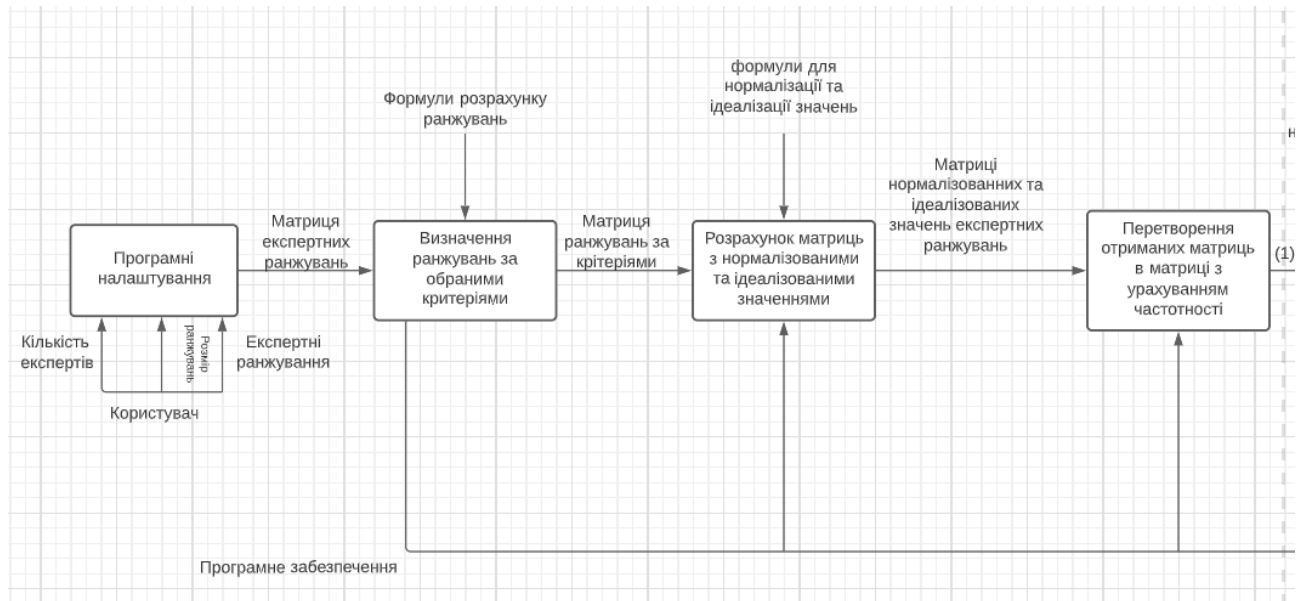


Рисунок 2.4 Деталізована IDEF0 перша частина

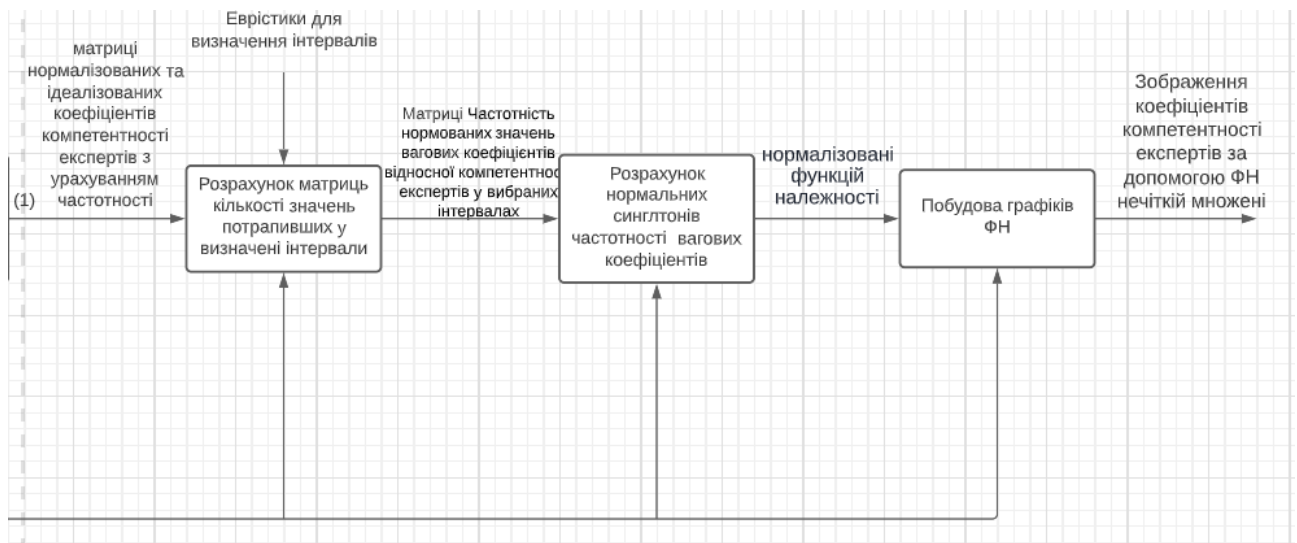


Рисунок 2.5 Деталізована IDEF0 друга частина

(1)-точка з'єднання малюнків

За цією діаграмою, першим кроком є отримання даних від користувача. Далі за отриманими значеннями розраховуються ранжування за обраними критеріями. Наступним кроком є розрахунок нормованих та ідеалізованих коефіцієнтів компетентності експертів які потім перетворюємо у матриці частотності коефіцієнтів відносної компетентності експертів у вибраних

інтервалах за допомогою яких далі ми можемо представити ці значення у вигляді нечітких значень. Останні кроки це нормування отриманих синглетонів та побудова графіків ФН нечіткій множині.

2.4 Розробка архітектури Система підтримки прийняття рішень при побудові функцій належності нечіткій множині

2.4.1 Розробка інтерфейсу користувача

Для написання коду було використано середовище Visual Studio Code та мову програмування Java Script.

Розробка інтерфейсу - це реалізація можливості введення кількості експертів, розміру ранжувань та нарешті самих експертних ранжувань для подальшої обробки. Користувач у вкладці браузера отримує запити на введення кількості ранжувань та їх розміру. Після збереження цієї інформації застосунок оброблює її та надає користувачеві порожню таблицю заданих розмірів для введення експертних ранжувань. Після заповнення даних користувач їх зберігає натиснувши на спеціальну кнопку.

2.4.2 Розробка коду для знаходження медіан за обраними методами

На цьому кроці я програмую математичні формули для знаходження медіан та обробляю їх для аналізу та подальшого використання. А саме розраховую найкращі ранжування за формулами:

Для метрики Кука (неспівпадання рангів об'єктів) при використанні утилітарного критерію обчислюється медіана Кука-Сейфорда

$$R^{CS} \in \Omega^{CS} = \text{Arg} \min_{R \in \Omega^R} \sum_{l \in L} d(R, R^l).$$

При використанні егалітарного критерію обчислюється ГВ-медіана (компроміс)

$$R^{FB} \in \Omega^{FB} = \text{Arg min}_{R \in \Omega^R} \max_{l \in L} d(R, R^l)$$

Для метрики Хемінга при використанні утилітарного критерію обчислюється медіана Кемені-Снелла:

$$R^{KC} \in \Omega^{KC} = \text{Arg min}_{B \in \Omega^B} \sum_{l \in L} d(B, B^l)$$

При використанні егалітарного критерію обчислюється ВГ-медіана (компроміс)

$$R^{BG} \in \Omega^{BG} = \text{Arg min}_{B \in \Omega^B} \max_{l \in L} d(B, B^l)$$

Тепер отримані медіани треба представити вигляди таблиці в якій будуть надані самі медіани, частотність їх обрання усіма методами, та відстані до експертних ранжувань.

2.4.3 Перетворення отриманих медіан у таблиці з ідеалізованими та нормованими значеннями вагових коефіцієнтів відносної компетентності експертів

Для того щоб отримати нормалізовані та ідеалізовані значення таблиці отриманої на попередньому етапі нам потрібно до кожного значення застосувати наступні формули:

Формула розрахунку нормованих значень

$$\gamma_{ij} = (1/d(R^i, R^j)) / \sum_{t=1}^s (1/d(R^t, R^j)), R^i \in \Omega^K, i=1, \dots, s, j \in I.$$

Формула розрахунку ідеалізованих значень

$$\gamma_{ij}^{id} = \gamma_{ij} / \max_{t \in I} \gamma_{it}, i=1, \dots, s, j \in I.$$

В результаті отримуємо дві таблиці: перша з нормованими значеннями; друга з ідеалізованими значеннями.

2.4.4 Перетворення таблиць з ідеалізованими та нормованими значеннями вагових коефіцієнтів відносної компетентності експертів в таблиці з урахуванням частотності.

В отриманих таблицях ідеалізованих та нормованих значень вагових коефіцієнтів відносної компетентності експертів треба позбутися від перших двох стовпців медіана та частотність. Для цього значень вагових коефіцієнтів ранжувань які мають частоту більше одиниці повторюються в відповідності до значення частоти. Стовпці ранжувань об'єктів та частоти заміню одним стовпцем з індексом.

2.4.5 Перетворення отриманих даних в ФН нечіткій множині

С початку розрахуємо синглтони частотності для отриманих значень. Для цього потрібно створити нову таблицю частотності нормованих значень вагових коефіцієнтів відносної компетентності експертів у вибраних інтервалах. Для нормованих значень крок буде 0.05 за евристиками E1 та E2, а для ідеалізованих значень 0.2 за евристиками E3 та E4. В таблиці буде зображено скільки значень потрапляє в визначений проміжок та за цими даними ми зможемо отримати синглтони частотності нормованих та ідеалізованих значень вагових коефіцієнтів.

2.4.6 Зображення отриманих нечітких значень у вигляді графіків ФН

Отримані синглтони частотності відображають кількість значень потрапивших у обраний інтервал для кожного з експертів. Для побудови графіків ФН потрібно нормувати отриманні значення. Для цього вважатиму інтервал в який попала максимальна кількість значень за той в якому ФН дорівнює 1 та розрахую коефіцієнт для розрахунку інших значень ФН. Помножимо кількість значень потрапивших в кожен інтервал на цей коефіцієнт та отримаю готові значення для побудови ФН.

Формула для розрахунку значень ФН

$x=k*i$, де x - значення ФН у заданному інтервалі; i – кількість значень потрапивших у заданий інтервал; k -коефіцієнт для розрахунку нормованих значень

$k=1/\max$, де k коефіцієнт для розрахунку нормованих значень, \max -максимальна кількість значень потрапивших в один інтервал.

2.5 Перспективи подальших досліджень

Очевидно, що експертів можна інтерпретувати як методи формування рейтингів об'єктів вимірювання чи оцінки чи інших джерел інформації на основі змісту розв'язуваної задачі, на якій базується ранжування нечислових об'єктів. Таким чином, можна досліджувати не тільки можливості, але й вагу, надійність, адекватність, зручність, надійність, швидкість обчислень, конвергенцію та інші параметри інформації, отриманої з різних джерел. Метод визначення коефіцієнта відносної компетентності експертів у задачі ранжування можна інтерпретувати як коефіцієнт відносної важливості джерел інформації [12]. Далі ми можемо розглядати формування та вирішення проблеми з точки зору її конкретної інтерпретації. Метод, описаний у роботі, можна покращити наступними способами:

- Сформулюйте питання про компетенцію експертів для визначення вільних рейтингів (Ідеальне квазі-впорядкування, сортування, квазі-послідовність, сортування з підключенням, квазі-впорядкування, сортування кластеризацією);

- Вивчити вплив ступеня згоди експертної групи на кількість ефективних рішень проблеми різних показників і різних критеріїв;

- Визначення експертного альянсу на основі кластеру експертного рейтингу великої кількості членів експертної групи;

- Дослідження не піддаються маніпуляції різними показниками та різними критеріями для визначення медіани даного рейтингу;

- Розрахувати вплив відносної «ваги» альянсів на результати рейтингу та чутливість показників і критеріїв до зміни «ваги» експертних альянсів;
- враховувати експертно встановлені коефіцієнти, що відображають вплив методу визначення медіани на остаточне розв'язання задачі: надійність методів, підвищення довіри до них шляхом дослідження або врахування експериментальних даних;
- вивчити необ'єктивність експертних висновків корупції та визначити ймовірність корупційних компонентів у вирішенні проблем експертної оцінки
- дослідити співвідношення між стандартизованими та ідеалізованими інтервалами коефіцієнтів відносної компетентності експертів;
- Застосування евристики [20] або генетичного алгоритму [21] для визначення медіани експертних рейтингів для пошуку медіани за наявності великої кількості об'єктів.

Розділ 3 Розробка програмного забезпечення Система підтримки прийняття рішень при побудові функцій належності нечіткій множині

3.1 Структура програмного забезпечення

Розглянемо структурну схему програмного забезпечення СППР (рис. 3.1).

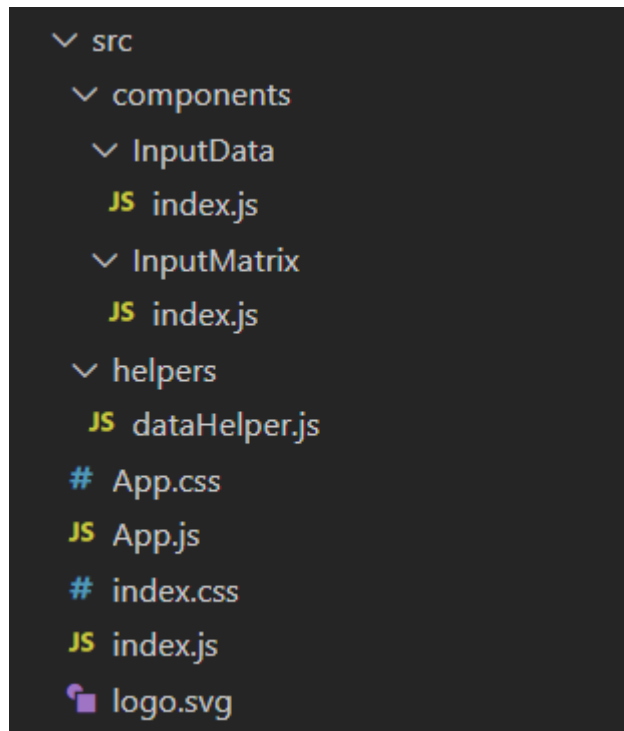


Рисунок 3.1 - Структурна схема програмних модулів

Специфікація програмних модулів

inputData index.js- компонент графического інтерфейса вводу даних

inputMatrix index.js- компонент графического інтерфейсу вводу ранжувачів

helpers dataHelper.js- код для проведення розрахунків на основі введених даних та повернення результату

app.css- опис стилів графічного інтерфейсу

app.js- реалізація та верстка графічного інтерфейсу

index.css- опис стилів головної сторінки

index.js- код ініціалізації верстки

logo.svg- іконка логотипу програми

3.2 Опис структури інтерфейсу користувача

Початкова сторінка містить два блоки для введення даних «Кількість ранжуваль» та «Розмір ранжуваль»

The image shows two identical input blocks. The first block has the label 'Кількість ранжуваль' followed by a text input field and a blue button labeled 'Зберігти'. The second block has the label 'Розмір ранжуваль' followed by a text input field and a blue button labeled 'Зберігти'.

Рисунок 3.2 Введення кількості ранжуваль та їх розміру

Після введення кількості ранжуваль та їх розміру з'являється таблиця заданої розмірності

The image shows a 6x5 grid of input fields. Each cell contains a number from 1 to 5. Below the grid is a blue button labeled 'Зберігти'.

4	1	3	2	5
3	2	4	5	1
1	4	3	5	2
2	5	1	4	3
5	1	3	4	2
2	3	4	1	5

Рисунок 3.3 Таблиця заповнена індивідуальними ранжуваннями

3.3 Опис результатів роботи застосунку

Після введення вхідних даних розроблений алгоритм розраховує медіани для заданої множини експертних ранжуваль у просторі усіх можливих ранжуваль. (рис. 3.4-3.7

[[1,2,3,5,4],[1,2,4,5,3],[1,3,4,5,2],[2,1,3,5,4],[2,1,4,5,3],[3,1,2,5,4]]

Сумарна відстань за метрикою Хемінга від кожної з обчислених медіан до 6 заданих є мінімальною і складає 50

Рисунок 3.4 Медіани Кемені-Снела для заданої множини експертних ранжувань у просторі усіх можливих ранжувань 5 об'єктів

[[1,2,3,5,4],[2,1,3,5,4]]

Максимальна відстань від цих медіан до усіх заданих дорівнює 10

Рисунок 3.5 Компромісні медіани, тобто ВГ-медіани

Медіан Кука-Сейфорда заданої множини 6 експертних ранжувань обчислено 5

[[2,1,3,4,5],[2,1,3,5,4],[2,1,4,5,3],[3,1,4,5,2],[4,1,3,5,2]]

Рисунок 3.6 Медіани Кука-Сейфорда заданої множини 6 експертних ранжувань

[[1,2,3,4,5],[1,2,3,5,4],[2,1,3,4,5],[2,1,3,5,4],[2,1,4,5,3],[2,1,5,4,3]]

Максимальна відстань від цих медіан до усіх заданих дорівнює 8

Рисунок 3.7 Компромісні медіани, тобто ГВ-медіани

В результаті розрахунку усіма методами отримуємо ранжування які задовольняють обраним критеріям. Далі треба для кожного ранжування розрахувати відстані до кожного з експертних ранжувань і в результаті отримаємо таблицю вигляду (рис 3.8)

РАНЖУВАННЯ	ЧАСТОТНІСТЬ	1	2	3	4	5	6
[1,2,3,5,4]	3	10	8	6	8	10	8
[1,2,4,5,3]	1	12	6	4	10	8	10
[1,3,4,5,2]	1	14	4	2	8	10	12
[2,1,3,5,4]	4	8	6	8	10	8	10
[2,1,4,5,3]	3	10	4	6	12	6	12
[3,1,2,5,4]	1	6	8	10	8	6	12
[2,1,3,4,5]	2	6	8	10	12	10	8
[3,1,4,5,2]	1	12	2	8	14	4	14
[4,1,3,5,2]	1	10	4	10	12	2	16
[1,2,3,4,5]	1	8	10	8	10	12	6
[2,1,5,4,3]	1	8	6	8	14	8	10

Рисунок 3.8 Результат обчислення медіан

Таблиця 1. Прикладні результати обчислення медіан , які задовольняють критеріям (3), (5), (7), (9)

Ранжування об'єктів	Частотність	Відстані від експертних ранжувань до медіан					
		1	2	3	4	5	6
{a1>a2>a3>a4>a5}	1	8	10	8	10	12	6
{a2>a1>a3>a4>a5}	2	6	8	10	12	10	8
{a1>a2>a3>a5>a4}	3	10	8	6	8	10	8
{a1>a2>a4>a5>a3}	1	12	6	4	10	8	10
{a1>a3>a4>a5>a2}	1	14	4	2	8	10	12
{a2>a1>a3>a5>a4}	4	8	6	8	10	8	10
{a2>a1>a4>a5>a3}	3	10	4	6	12	6	12
{a3>a1>a2>a5>a4}	1	6	8	10	8	6	12
{a3>a1>a4>a5>a2}	1	12	2	8	14	4	14
{a4>a1>a3>a5>a2}	1	10	4	10	12	2	16
{a2>a1>a5>a4>a3}	1	8	6	8	14	8	10

Результати співпадають з обчисленнями.

На наступну кроці розраховуємо дві нові таблиці:

РАНЖУВАННЯ	ЧАСТОТНІСТЬ	1	2	3	4	5	6
[1,2,3,5,4]	3	0.135	0.169	0.225	0.169	0.135	0.169
[1,2,4,5,3]	1	0.101	0.202	0.303	0.121	0.152	0.121
[1,3,4,5,2]	1	0.063	0.221	0.443	0.111	0.089	0.074
[2,1,3,5,4]	4	0.169	0.225	0.169	0.135	0.169	0.135
[2,1,4,5,3]	3	0.118	0.294	0.196	0.098	0.196	0.098
[3,1,2,5,4]	1	0.217	0.163	0.13	0.163	0.217	0.109
[2,1,3,4,5]	2	0.238	0.179	0.143	0.119	0.143	0.179
[3,1,4,5,2]	1	0.076	0.454	0.114	0.065	0.227	0.065
[4,1,3,5,2]	1	0.091	0.228	0.091	0.076	0.456	0.057
[1,2,3,4,5]	1	0.179	0.143	0.179	0.143	0.119	0.238
[2,1,5,4,3]	1	0.175	0.234	0.175	0.1	0.175	0.14

Рисунок 3.9 Нормовані значення вагових коефіцієнтів відносної компетентності експертів залежно від того, яка з медіан є розв'язком задачі визначення групового ранжування

РАНЖУВАННЯ	ЧАСТОТНІСТЬ	1	2	3	4	5	6
[1,2,3,5,4]	3	0.6	0.75	1	0.75	0.6	0.75
[1,2,4,5,3]	1	0.33	0.67	1	0.4	0.5	0.4
[1,3,4,5,2]	1	0.14	0.5	1	0.25	0.2	0.17
[2,1,3,5,4]	4	0.75	1	0.75	0.6	0.75	0.6
[2,1,4,5,3]	3	0.4	1	0.67	0.33	0.67	0.33
[3,1,2,5,4]	1	1	0.75	0.6	0.75	1	0.5
[2,1,3,4,5]	2	1	0.75	0.6	0.5	0.6	0.75
[3,1,4,5,2]	1	0.17	1	0.25	0.14	0.5	0.14
[4,1,3,5,2]	1	0.2	0.5	0.2	0.17	1	0.13
[1,2,3,4,5]	1	0.75	0.6	0.75	0.6	0.5	1
[2,1,5,4,3]	1	0.75	1	0.75	0.43	0.75	0.6

Рисунок 3.10 Ідеалізовані значення вагових коефіцієнтів відносної компетентності експертів

Тепер за отриманими значеннями перетворюємо отримані таблиці в таблиці з урахуванням частотності значень:

№	1	2	3	4	5	6
1	0.063	0.143	0.091	0.065	0.089	0.057
2	0.076	0.163	0.114	0.076	0.119	0.065
3	0.091	0.169	0.13	0.098	0.135	0.074
4	0.101	0.169	0.143	0.098	0.135	0.098
5	0.118	0.169	0.143	0.098	0.135	0.098
6	0.118	0.179	0.169	0.1	0.143	0.098
7	0.118	0.179	0.169	0.111	0.143	0.109
8	0.135	0.202	0.169	0.119	0.152	0.121
9	0.135	0.221	0.169	0.119	0.169	0.135
10	0.135	0.225	0.175	0.121	0.169	0.135
11	0.169	0.225	0.179	0.135	0.169	0.135
12	0.169	0.225	0.196	0.135	0.169	0.135
13	0.169	0.225	0.196	0.135	0.175	0.14
14	0.169	0.228	0.196	0.135	0.196	0.169
15	0.175	0.234	0.225	0.143	0.196	0.169
16	0.179	0.294	0.225	0.163	0.196	0.169
17	0.217	0.294	0.225	0.169	0.217	0.179
18	0.238	0.294	0.303	0.169	0.227	0.179
19	0.238	0.454	0.443	0.169	0.456	0.238

Рисунок 3.11 Нормовані значення вагових коефіцієнтів відносної компетентності експертів з урахуванням частотності значень

№	1	2	3	4	5	6
1	0.14	0.5	0.2	0.14	0.2	0.13
2	0.17	0.5	0.25	0.17	0.5	0.14
3	0.2	0.6	0.6	0.25	0.5	0.17
4	0.33	0.67	0.6	0.33	0.5	0.33
5	0.4	0.75	0.6	0.33	0.6	0.33
6	0.4	0.75	0.67	0.33	0.6	0.33
7	0.4	0.75	0.67	0.4	0.6	0.4
8	0.6	0.75	0.67	0.43	0.6	0.5
9	0.6	0.75	0.75	0.5	0.6	0.6
10	0.6	0.75	0.75	0.5	0.67	0.6
11	0.75	1	0.75	0.6	0.67	0.6
12	0.75	1	0.75	0.6	0.67	0.6
13	0.75	1	0.75	0.6	0.75	0.6
14	0.75	1	0.75	0.6	0.75	0.75
15	0.75	1	1	0.6	0.75	0.75
16	0.75	1	1	0.75	0.75	0.75
17	1	1	1	0.75	0.75	0.75
18	1	1	1	0.75	1	0.75
19	1	1	1	0.75	1	1

Рисунок 3.12 Ідеалізовані значення вагових коефіцієнтів відносної компетентності експертів з урахуванням частотності значень

Обчислимо ці таблиці самостійно щоб перевірити працездатність коду:

Таблиця 2. Нормовані значення вагових коефіцієнтів відносної компетентності експертів з урахуванням частотності значень

Номер п/п	Нормовані вагові коефіцієнти компетентності					
	1	2	3	4	5	6
1	0,063	0,143	0,091	0,065	0,089	0,057
2	0,076	0,163	0,114	0,076	0,119	0,065
3	0,091	0,169	0,130	0,098	0,135	0,074
4	0,101	0,169	0,143	0,098	0,135	0,098
5	0,118	0,169	0,143	0,098	0,135	0,098
6	0,118	0,179	0,169	0,100	0,143	0,098
7	0,118	0,179	0,169	0,111	0,143	0,109
8	0,135	0,202	0,169	0,119	0,152	0,121
9	0,135	0,221	0,169	0,119	0,169	0,135
10	0,135	0,225	0,175	0,121	0,169	0,135
11	0,169	0,225	0,179	0,135	0,169	0,135
12	0,169	0,225	0,196	0,135	0,169	0,135
13	0,169	0,225	0,196	0,135	0,175	0,140
14	0,169	0,228	0,196	0,135	0,196	0,169
15	0,175	0,234	0,225	0,143	0,196	0,169
16	0,179	0,294	0,225	0,163	0,196	0,169
17	0,217	0,294	0,225	0,169	0,217	0,179
18	0,238	0,294	0,303	0,169	0,227	0,179
19	0,238	0,454	0,443	0,169	0,456	0,238

Таблиця 3. Ідеалізовані значення вагових коефіцієнтів відносної компетентності експертів з урахуванням частотності значень

Номер п/п	Ідеалізовані вагові коефіцієнти компетентності					
	1	2	3	4	5	6
1	0,143	0,500	0,200	0,143	0,200	0,125
2	0,167	0,500	0,250	0,167	0,500	0,143
3	0,200	0,600	0,600	0,250	0,500	0,167
4	0,333	0,667	0,600	0,333	0,500	0,333
5	0,400	0,750	0,600	0,333	0,600	0,333
6	0,400	0,750	0,667	0,333	0,600	0,333
7	0,400	0,750	0,667	0,400	0,600	0,400
8	0,600	0,750	0,667	0,429	0,600	0,500
9	0,600	0,750	0,750	0,500	0,600	0,600
10	0,600	0,750	0,750	0,500	0,667	0,600
11	0,750	1,000	0,750	0,600	0,667	0,600
12	0,750	1,000	0,750	0,600	0,667	0,600
13	0,750	1,000	0,750	0,600	0,750	0,600
14	0,750	1,000	0,750	0,600	0,750	0,750
15	0,750	1,000	1,000	0,600	0,750	0,750
16	0,750	1,000	1,000	0,750	0,750	0,750
17	1,000	1,000	1,000	0,750	0,750	0,750
18	1,000	1,000	1,000	0,750	1,000	0,750
19	1,000	1,000	1,000	0,750	1,000	1,000

Результати співпадають з цього слідує що код працює вірно.

Далі ми розраховуємо частотність нормованих значень вагових коефіцієнтів відносної компетентності експертів у вибраних інтервалах.

За евристиками E1 та E2 крок між інтервалами 0.05, мінімальне значення 0.063, максимальне 0.456, тоді наш проміжок [0.05;0.5] (рис. 3.9)

ЗНАЧЕННЯ	1	2	3	4	5	6
0	0	0	0	0	0	0
0.05	0	0	0	0	0	0
0.1	3	0	1	6	1	6
0.15	7	1	4	9	6	7
0.2	6	6	9	4	9	5
0.25	3	8	3	0	2	1
0.3	0	3	0	0	0	0
0.35	0	0	1	0	0	0
0.4	0	0	0	0	0	0
0.45	0	0	1	0	0	0
0.5	0	1	0	0	1	0

Рисунок 3.13 Частотність нормованих значень вагових коефіцієнтів відносної компетентності експертів у вибраних інтервалах

За частотністю в заданих проміжках отримуємо значення синглотонів частотності нормованих значень вагових коефіцієнтів

$$\mu_1(0.1)/3 + \mu_1(0.15)/7 + \mu_1(0.2)/6 + \mu_1(0.25)/3$$

$$\mu_2(0.15)/1 + \mu_2(0.2)/6 + \mu_2(0.25)/8 + \mu_2(0.3)/3 + \mu_2(0.5)/1$$

$$\mu_3(0.1)/1 + \mu_3(0.15)/4 + \mu_3(0.2)/9 + \mu_3(0.25)/3 + \mu_3(0.35)/1 + \mu_3(0.45)/1$$

$$\mu_4(0.1)/6 + \mu_4(0.15)/9 + \mu_4(0.2)/4$$

$$\mu_5(0.1)/1 + \mu_5(0.15)/6 + \mu_5(0.2)/9 + \mu_5(0.25)/2 + \mu_5(0.5)/1$$

$$\mu_6(0.1)/6 + \mu_6(0.15)/7 + \mu_6(0.2)/5 + \mu_6(0.25)/1$$

Рисунок 3.14 Синглтони частотності нормованих значень

Перевіримо достовірність отриманих значень розрахувавши кількість значень потрапивших у обрані інтервали.

Таблиця 4. Частотність нормованих значень вагових коефіцієнтів відносної компетентності експертів у вибраних інтервалах

Значення, що представляє вибраний	Кількість значень нормованих вагових коефіцієнти компетентності експертів, які
-----------------------------------	--

інтервал	потрапляють у вибраний інтервал					
	1	2	3	4	5	6
0,1	3		1	6	1	6
0,15	7	1	4	9	6	7
0,2	6	6	9	4	9	5
0,25	3	8	3		2	1
0,3		3				
0,35			1			
0,4						
0,45			1			
0,5		1			1	

Як ми можемо побачити результати співпадають це свідчить про коректну роботу програмного застосунку.

Далі здійснимо нормалізацію функцій належності і представимо нечіткі значення відносних коефіцієнтів компетентності експертів у вигляді нормальних функцій належності

$$\mu_1(0.1)/0.43 + \mu_1(0.15)/1 + \mu_1(0.2)/0.86 + \mu_1(0.25)/0.43$$

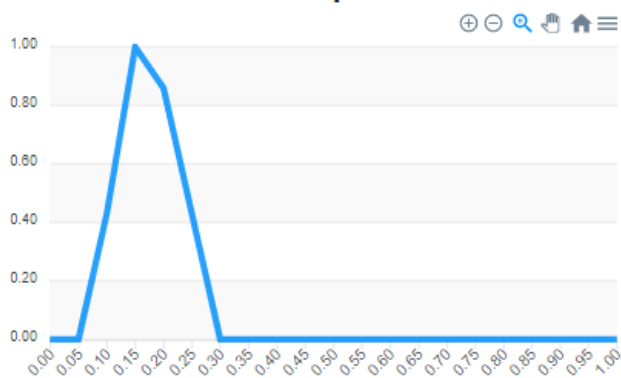


Рисунок 3.15 ФН для експерта 1

$$\mu_2(0.15)/0.13 + \mu_2(0.2)/0.75 + \mu_2(0.25)/1 + \mu_2(0.3)/0.38 + \mu_2(0.5)/0.13$$

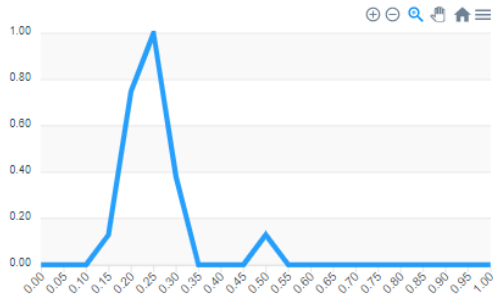


Рисунок 3.16 ФН для эксперта 2

$$\mu_3(0.1)/0.11 + \mu_3(0.15)/0.44 + \mu_3(0.2)/1 + \mu_3(0.25)/0.33 + \mu_3(0.35)/0.11 + \mu_3(0.45)/0.11$$

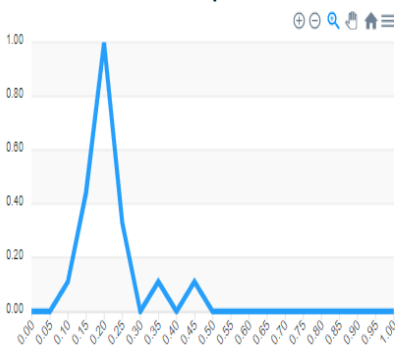


Рисунок 3.17 ФН для эксперта 3

$$\mu_4(0.1)/0.67 + \mu_4(0.15)/1 + \mu_4(0.2)/0.44$$

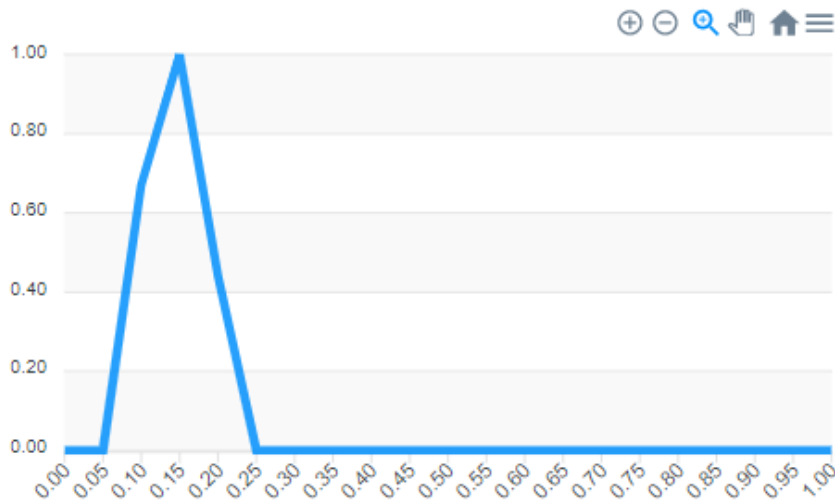


Рисунок 3.18 ФН для эксперта 4

$$\mu_5(0.1)/0.11 + \mu_5(0.15)/0.67 + \mu_5(0.2)/1 + \mu_5(0.25)/0.22 + \mu_5(0.5)/0.11$$

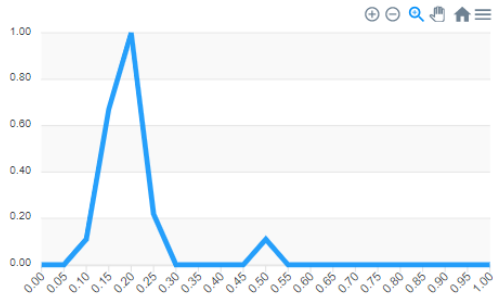


Рисунок 3.15 ФН для експерта 5

$$\mu_6(0.1)/0.86 + \mu_6(0.15)/1 + \mu_6(0.2)/0.71 + \mu_6(0.25)/0.14$$

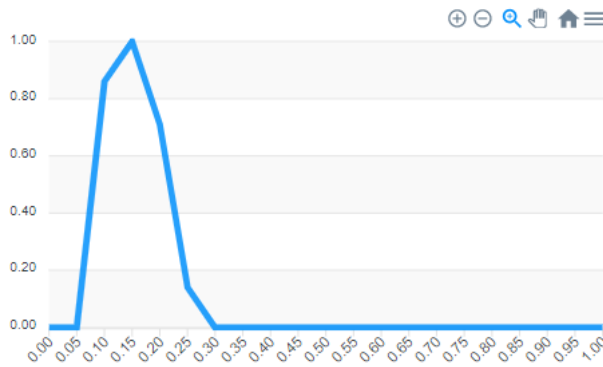


Рисунок 3.19 ФН для експерта 6

Та за евристикими Е3 та Е4 для ідеалізованих значень отримаємо проміжки для ідеалізованих значень в проміжку [0;1] з кроком 0.2. (рис. 3.16)

ЗНАЧЕННЯ	1	2	3	4	5	6
0	0	0	0	0	0	0
0.2	3	0	1	2	1	3
0.4	4	0	1	5	0	4
0.6	3	3	3	8	8	6
0.8	6	7	9	4	8	5
1	3	9	5	0	2	1

Рисунок 3.20 Частотність ідеалізованих значень вагових коефіцієнтів відносної компетентності експертів у вибраних інтервалах

За частотністю в заданих проміжках отримуємо значення синглотонів частотності ідеалізованих значень вагових коефіцієнтів

$$\mu_1(0.2)/3 + \mu_1(0.4)/4 + \mu_1(0.6)/3 + \mu_1(0.8)/6 + \mu_1(1)/3$$

$$\mu_2(0.6)/3 + \mu_2(0.8)/7 + \mu_2(1)/9$$

$$\mu_3(0.2)/1 + \mu_3(0.4)/1 + \mu_3(0.6)/3 + \mu_3(0.8)/9 + \mu_3(1)/5$$

$$\mu_4(0.2)/2 + \mu_4(0.4)/5 + \mu_4(0.6)/8 + \mu_4(0.8)/4$$

$$\mu_5(0.2)/1 + \mu_5(0.6)/8 + \mu_5(0.8)/8 + \mu_5(1)/2$$

$$\mu_6(0.2)/3 + \mu_6(0.4)/4 + \mu_6(0.6)/6 + \mu_6(0.8)/5 + \mu_6(1)/1$$

Рисунок 3.21 Синглтони частотності ідеалізованих значень

Знов перевіримо достовірність отриманих значень розрахувавши кількість значень потрапивших у обрані інтервали.

Таблиця 5. Частотність ідеалізованих значень вагових коефіцієнтів відносної компетентності експертів у вибраних інтервалах

Значення, що представляє вибраний інтервал	Кількість значень ідеалізованих вагових коефіцієнти компетентності експертів, які потрапляють у вибраний інтервал					
	1	2	3	4	5	6
0,2	3	0	1	2	1	3
0,4	4	0	1	5	0	4
0,6	3	3	3	8	8	6
0,8	6	7	9	4	8	5
1	3	9	5	0	2	1

Результаті співпадають з програмними розрахунками.

Знов здійснимо нормалізацію функцій належності і представимо нечіткі значення відносних коефіцієнтів компетентності експертів у вигляді нормальних функцій належності

$$\mu_1(0.2)/0.5 + \mu_1(0.4)/0.67 + \mu_1(0.6)/0.5 + \mu_1(0.8)/1 + \mu_1(1)/0.5$$

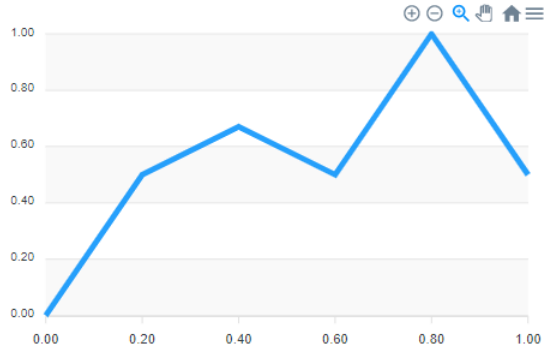


Рисунок 3.22 ФН для експерта 1

$$\mu_2(0.6)/0.33 + \mu_2(0.8)/0.78 + \mu_2(1)/1$$

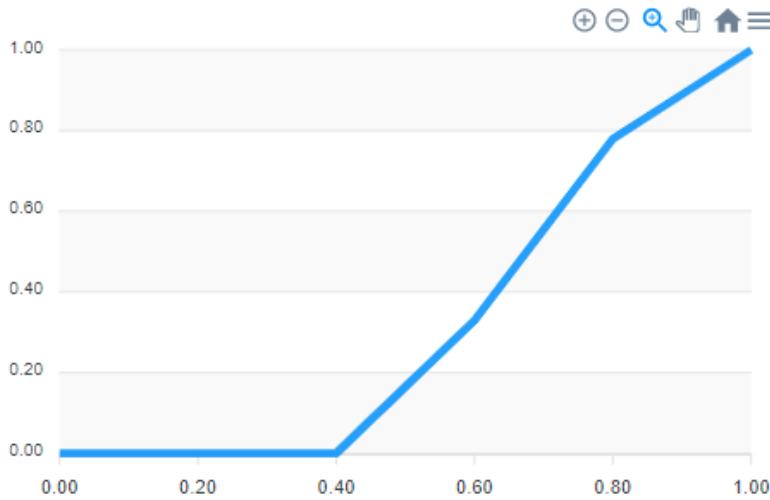


Рисунок 3.23 ФН для експерта 2

$$\mu_3(0.2)/0.11 + \mu_3(0.4)/0.11 + \mu_3(0.6)/0.33 + \mu_3(0.8)/1 + \mu_3(1)/0.56$$

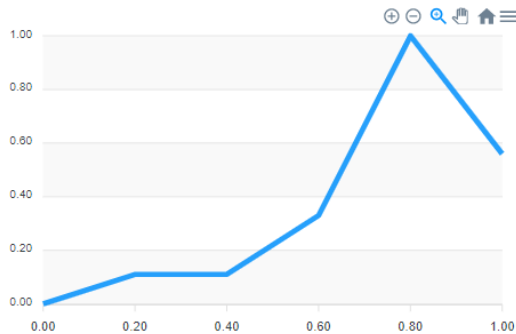


Рисунок 3.24 ФН для эксперта 3

$$\mu_4(0.2)/0.25 + \mu_4(0.4)/0.63 + \mu_4(0.6)/1 + \mu_4(0.8)/0.5$$

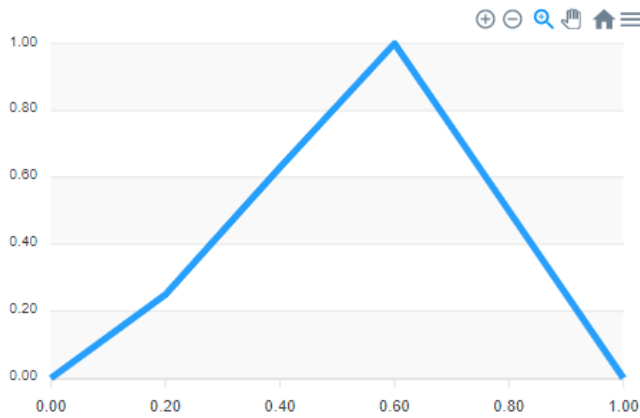


Рисунок 3.25 ФН для эксперта 4

$$\mu_5(0.2)/0.13 + \mu_5(0.6)/1 + \mu_5(0.8)/1 + \mu_5(1)/0.25$$

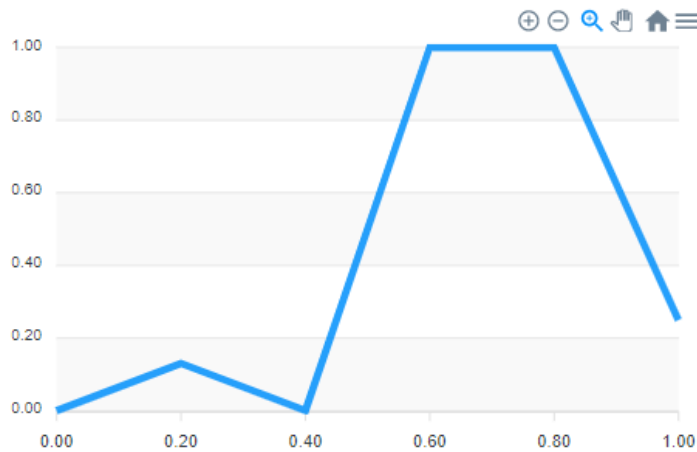


Рисунок 3.26 ФН для эксперта 5

$$\mu_6(0.2)/0.5 + \mu_6(0.4)/0.67 + \mu_6(0.6)/1 + \mu_6(0.8)/0.83 + \mu_6(1)/0.17$$

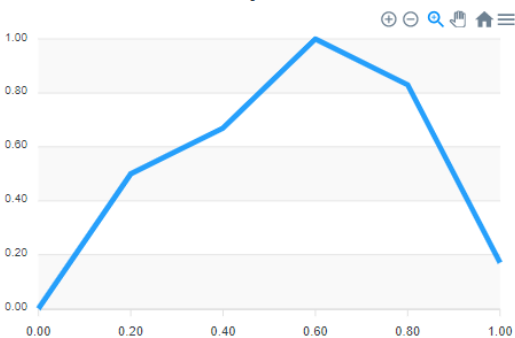


Рисунок 3.27 ФН для эксперта 6

Як ми можемо побачити ФН приймає найбільше значення саме в тому інтервалі куди потрапляє найбільша кількість точок по цьому експерту. Через те що ми нормалізуємо ФН ми отримуємо значення максимуму дорівнююче 1 і саме у цій точці обраний експерт матиме максимальне значення коефіцієнту компетентності. Наприклад 1й експерт матиме максимальне значення коефіцієнту компетентності у точці 0.15, 2й у точці 0.25 і т.д. За отриманими графіками ще можна дослідити поведінку коефіцієнту компетентності в околицях точок.

Висновок

В результаті роботи було розроблено застосунок який допомагає визначити відносну компетентність експерта на заданих проміжках. Визначив точки в яких максимальні значення коефіцієнта компетентності для кожного експерта, та поведінку коефіцієнтів компетентності експертів в околицях цих точок. Це є перевагою запропонованого методу порівняно з методами визначення фіксованих коефіцієнтів компетентності або інтервальних коефіцієнтів. Я створив інструмент для подальшого використання при розв'язанні багатокритеріальних задач, задач підтримки прийняття рішень, задач обробки експертної інформації.

Джерела

1. Волошин О.Ф., Мащенко С.О. Теорія прийняття рішень: Навчальний посібник. – К.: Видавничо-поліграфічний центр "Київський університет", 2006. – 304 с.
2. Орлов А.И. Методы принятия управленческих решений: учебник – М.: КНОРУС, 2018. – 286 с.
3. Гнатієнко Г.М., Снитюк В.Є. Експертні технології прийняття рішень: Монографія. – К.: ТОВ «Маклаут», 2008. – 444с.
4. Саати Т.Л. Относительное измерение и его обобщение в принятии решений. Почему парные сравнения являются ключевыми в математике для измерения неосознанных факторов / Т.Л. Саати // Cloud of Science. – 2016. – Т. 3, № 2. – С. 171–262.
5. За ред. В.Є. Снитюка Міжнародний науковий симпозіум «ІНТЕЛЕКТУАЛЬНІ РІШЕННЯ» ОБЧИСЛЮВАЛЬНИЙ ІНТЕЛЕКТ (РЕЗУЛЬТАТИ, ПРОБЛЕМИ, ПЕРСПЕКТИВИ) с 23-26
6. В. Г. Тоценко Методи та засоби підтримки прийняття рішень <https://uchika.in.ua/metodi-ta-zasobi-pidtrimki-prijnyattya-rishene.html>
7. Г.М. Гнатієнко, В.Є. Снитюк ЕКСПЕРТНІ ТЕХНОЛОГІЇ ПРИЙНЯТТЯ РІШЕНЬ
8. Теорія прийняття технічних рішень https://ab.uu.edu.ua/edu-discipline/teoriya_priinyattya_tekhnichnikh_rishen
9. АПОСТЕРІОРНЕ ВИЗНАЧЕННЯ КОМПЕТЕНТНОСТІ ЕКСПЕРТІВ В УМОВАХ НЕВИЗНАЧЕНОСТІ Гнатієнко Г.М., Снитюк В.Є.
- 10 Снитюк В.Є., Рифат Мохаммед Али, Моделі и методи визначення компетентности експертів на основі аксиоми неземіщенності// Вісник Черкаського інженерно-технологічного інституту. – 2000. – т 4. – С. 121–126.
11. Ю.В. Мирончук, О.М. Купріненко ПОБУДОВА ФУНКЦІЙ НАЛЕЖНОСТІ НЕЧІТКИХ МНОЖИН, ЯКІ ВІДПОВІДАЮТЬ КІЛЬКІСНИМ ЕКСПЕРТНИМ ОЦІНКАМ ФІЗИЧНИХ ВЕЛИЧИН
12. Додонов А.Г., Ландэ Д.В., Цыганок В.В., Андрейчук О.В., Каденко С.В, Грайворонская А.Н. Распознавание информационных операций. – К.: ООО «Инжиниринг», 2017. – 282 с.
13. Макаров И. М., Виноградская Т. М., Рубчинский А. А., Соколов В. В. Теория выбора и принятия решений. М.: Наука, 1982. 330 с.
14. Миркин Б.Г. Проблема группового выбора. М.: Наука, 1973. – 256 с.
15. Холлендер М., Вулф Д. Непараметрические методы статистики. М.: Финансы и статистика, 1983. – 518 с.

16. Гнатієнко Г.М. Алгоритми визначення функції належності шляхом аналізу частотності значень//Праці III-ї міжнародної школи-семінару «Теорія прийняття рішень», Ужгород, УжНУ, 2006.-С.32-34.

17. Саати Т.Л. Относительное измерение и его обобщение в принятии решений. Почему парные сравнения являются ключевыми в математике для измерения неосязаемых факторов / Т.Л. Саати // Cloud of Science. – 2016. – Т. 3, № 2. – С. 171–262.

18. Саати Т.Л. Об измерении неосязаемого. Подход к относительным измерениям на основе главного собственного вектора матрицы парных сравнений / Т.Л. Саати // Cloud of Science. – 2015. – Т. 2, № 1. – С. 5–39.

19. Гнатієнко Г.М. Визначення вагових коефіцієнтів критеріїв задачі багатокритеріальної оптимізації у вигляді функцій належності нечіткій множині // Матеріали доповідей V Міжнародної науково практичної конференції «Інформаційні технології та взаємодії» (IT&I – 2018). – К. ВПЦ «Київський університет», 2018. – 343 с. С.29-30.

20. Hnatiienko H.M., Hnatiienko V.H. Heuristic algorithm for determining compromise rankings on a set of individual expert rankings // INTELLIGENT SOLUTIONS. Decision Making Theory: Proceedings of the International School-Seminar, April 15-20, 2019, Uzhhorod,Ukraine / Ministry of Education and Science of Ukraine, Uzhhorod national university and [etc]; Leonid F. Hulianytskyi (Editor). – Pp.53-54.

21. Hnatiienko H.M., Kruglov A.I. Definition of a compromise ranking on the set of individual rankings using the genetic algorithm // Міжнародний науковий симпозіум «ІНТЕЛЕКТУАЛЬНІ РІШЕННЯ». Обчислювальний інтелект (результати, проблеми, перспективи): праці міжнар. наук.-практ. конф., 15-20 квітня 2019 р., Ужгород / М-во освіти і науки України, ДВНЗ «Ужгородський національний університет», та [ін.]; наук. Ред. В.Є. Снитюк. С.63-64

Додатки

Додаток 1

```

const arrayEquals = (a, b) =>
  Array.isArray(a) &&
  Array.isArray(b) &&
  a.length === b.length &&
  a.every((val, index) => val === b[index]);

const kukDistance = (a, b) =>
  a.reduce((acc, v, i) => acc + Math.abs(v - b[i]), 0);

const hemingDistance = (a, b) => {
  let sum = 0;
  for (let i = 0; i < a.length; i++) {
    for (let j = 0; j < a[i].length; j++) {
      sum += Math.abs(a[i][j] - b[i][j]);
    }
  }
  return sum;
};

const mpp = (a) => {
  const res = [];
  for (let w = 0; w < a.length; w++) res.push([]);
  for (let i = 0; i < a.length; i++) {
    for (let j = 0; j < a.length; j++) {
      res[i][j] = a[i] >= a[j] ? 1 : 0;
    }
  }
  return res;
};

const permutator = (inputArr) => {
  let result = [];

  const permute = (arr, m = []) => {
    if (arr.length === 0) {
      result.push(m);
    } else {
      for (let i = 0; i < arr.length; i++) {
        let curr = arr.slice();
        let next = curr.splice(i, 1);
        permute(curr.slice(), m.concat(next));
      }
    }
  };

  permute(inputArr);

  return result;
};

const findMin = (a) => a.reduce((acc, el) => Math.min(acc, el), a[0]);
const findMax = (a) => a.reduce((acc, el) => Math.max(acc, el), a[0]);

```

```

const getMinIds = (a) => {
  const min = findMin(a);
  return a.map((el, i) => (el == min ? i : false)).filter((e) => e !== false);
};

const getDistsFromOneMedianToOthers = (distFunc, med, q) => {
  const res = [];
  for (let i = 0; i < q.length; i++) {
    res[i] = distFunc(med, q[i]);
  }
  return res;
};

const getMinElements = (arr, q) => getMinIds(arr).map((i) => q[i]);

const getKey = (a) => `[${a.join(",")}]`;
const getFromKey = (a) =>
  a
    .slice(1, -1)
    .split(",")
    .map((e) => +e);

const getRange = (step) => {
  const res = [];
  for (let i = 0; i <= 1.05; i += step) {
    res.push(Math.round(i * 100) / 100);
  }
  return res;
};

export const start = (q) => {
  const result = [];
  const pp = permutator([...Array(q[0].length + 1).keys()].slice(1));
  const p = [];
  for (let i = 0; i < pp.length; i++) {
    let flag = true;
    for (let j = 0; j < q.length; j++) {
      if (arrayEquals(q[j], pp[i])) {
        flag = false;
        break;
      }
    }
    if (flag) {
      p.push(pp[i]);
    }
  }

  const medians = {};
  // ----- 7 9 -----
  let sums = new Array(p.length).fill(0);

  for (let i = 0; i < p.length; i++) {
    for (let j = 0; j < q.length; j++) {
      sums[i] += hemingDistance(mpp(p[i]), mpp(q[j]));
    }
  }
}

```

```

const hemMin = findMin(sums);
const hemMedians = sums
  .map((el, i) => (el == hemMin ? i : 0))
  .filter((e) => e)
  .map((el) => p[el]);
result.push(
  `Медіан Кемені-Снела для заданої множини експертних ранжувань у просторі усіх
  можливих ранжувань ${q[0].length} об'єктів обчислено ${hemMedians.length}`
);
result.push(hemMedians);
hemMedians.forEach((m) => {
  const k = getKey(m);
  if (!medians[k]) {
    medians[k] = 1;
  } else {
    medians[k]++;
  }
});
result.push(
  `Сумарна відстань за метрикою Хемінга від кожної з обчислених медіан до
  ${q.length} заданих є мінімальною і складає ${hemMin}`
);

let hemMins = new Array(p.length).fill(0);

for (let i = 0; i < p.length; i++) {
  for (let j = 0; j < q.length; j++) {
    hemMins[i] = Math.max(hemMins[i], hemingDistance(mpp(p[i]), mpp(q[j])));
  }
}

result.push(
  `Компромісних медіан, тобто ВГ-медіан обчислено ${
    getMinElements(hemMins, p).length
  }`
);
const hemMinsMedians = getMinElements(hemMins, p);
result.push(hemMinsMedians);

hemMinsMedians.forEach((m) => {
  const k = getKey(m);
  if (!medians[k]) {
    medians[k] = 1;
  } else {
    medians[k]++;
  }
});
result.push(
  `Максимальна відстань від цих медіан до усіх заданих дорівнює ${findMin(
    hemMins
  )}`
);

// ----- 3 5 -----

sums = new Array(p.length).fill(0);

```

```

for (let i = 0; i < p.length; i++) {
  for (let j = 0; j < q.length; j++) {
    if (JSON.stringify(p[i]) !== JSON.stringify(q[j]))
      sums[i] += kukDistance(p[i], q[j]);
  }
}

const kukMin = findMin(sums);
const kukMedians = sums
  .map((el, i) => (el == kukMin ? i : 0))
  .filter((e) => e)
  .map((el) => p[el]);
result.push(
  `Медіан Кука-Сейфорда заданої множини ${q.length} експертних ранжувань
  обчислено ${kukMedians.length}`
);
result.push(kukMedians);
kukMedians.forEach((m) => {
  const k = getKey(m);
  if (!medians[k]) {
    medians[k] = 1;
  } else {
    medians[k]++;
  }
});
result.push(
  `Сумарна відстань за метрикою Кука неспівпадання рангів від кожної з обчислених
  медіан до ${q.length} заданих є мінімальною і складає ${kukMin}`
);

let kukMins = new Array(p.length).fill(0);

for (let i = 0; i < p.length; i++) {
  for (let j = 0; j < q.length; j++) {
    kukMins[i] = Math.max(kukMins[i], kukDistance(p[i], q[j]));
  }
}

const kukMinMedians = getMinElements(kukMins, p);
result.push(
  `Компромісних медіан, тобто ГВ-медіан обчислено ${kukMinMedians.length}`
);

result.push(kukMinMedians);
kukMinMedians.forEach((m) => {
  const k = getKey(m);
  if (!medians[k]) {
    medians[k] = 1;
  } else {
    medians[k]++;
  }
});
result.push(
  `Максимальна відстань від цих медіан до усіх заданих дорівнює ${findMin(
    kukMins
  )}`
);

```

```

const arrs = {};

result.push(
  "Результати обчислення медіан, які задовольняють критеріям (3), (5), (7), (9)"
);
let f = [];
Object.entries(medians).forEach(([k, v]) => {
  arrs[k] = [];
  arrs[k][0] = getDistsFromOneMedianToAnothers(
    hemingDistance,
    mpp(getFromKey(k)),
    q.map((e) => mpp(e))
  );
  f.push([k, v, arrs[k][0]]);
});
result.push(f);

result.push(
  "Нормовані значення вагових коефіцієнтів відносної компетентності експертів залежно від того, яка з медіан є розв'язком задачі визначення групового ранжування"
);
f = [];
Object.entries(medians).forEach(([k, v]) => {
  const dists = getDistsFromOneMedianToAnothers(
    hemingDistance,
    mpp(getFromKey(k)),
    q.map((e) => mpp(e))
  );
  const sum = dists.reduce((acc, v) => acc + 1 / v, 0);
  arrs[k][1] = dists.map((e1) => Math.round((1 / e1 / sum) * 1000) / 1000);
  f.push([k, v, arrs[k][1]]);
});
result.push(f);

f = [];
result.push(
  "Ідеалізовані значення вагових коефіцієнтів відносної компетентності експертів"
);
Object.entries(medians).forEach(([k, v]) => {
  const max = findMax(arrs[k][1]) || 1;
  arrs[k][2] = arrs[k][1].map((e1) => Math.round((e1 / max) * 100) / 100);
  f.push([k, v, arrs[k][2]]);
});
result.push(f);

result.push(
  "Нормовані значення вагових коефіцієнтів відносної компетентності експертів з урахуванням частотності значень"
);
let table1 = new Array(q.length);
for (let i = 0; i < table1.length; i++) {
  table1[i] = [];
}
Object.entries(medians).forEach(([k, v]) => {
  for (let i = 0; i < arrs[k][1].length; i++) {

```

```

    for (let j = 0; j < v; j++) {
      table1[i].push(arrs[k][1][i]);
    }
  }
});
table1 = table1.map((e) => e.sort());
result.push(table1);
result.push(
  "Ідеалізовані значення вагових коефіцієнтів відносної компетентності експертів
з урахуванням частотності значень"
);
let table2 = new Array(q.length);
for (let i = 0; i < table2.length; i++) {
  table2[i] = [];
}
Object.entries(medians).forEach(([k, v]) => {
  for (let i = 0; i < arrs[k][2].length; i++) {
    for (let j = 0; j < v; j++) {
      table2[i].push(arrs[k][2][i]);
    }
  }
});
table2 = table2.map((e) => e.sort());
result.push(table2);

result.push(
  "Частотність нормованих значень вагових коефіцієнтів відносної компетентності
експертів у вибраних інтервалах"
);
let table3 = new Array(q.length);
const range1 = getRange(0.05);
for (let i = 0; i < table3.length; i++) {
  table3[i] = {};
  for (let u = 0; u < range1.length; u++) {
    table3[i][range1[u]] = 0;
  }
}

for (let i = 0; i < table1.length; i++) {
  for (let j = 0; j < table1[i].length; j++) {
    for (let k = 0; k < range1.length - 1; k++) {
      if (table1[i][j] > range1[k - 1] && table1[i][j] <= range1[k]) {
        if (!table3[i][range1[k]]) {
          table3[i][range1[k]] = 1;
        } else {
          table3[i][range1[k]]++;
        }
      }
    }
  }
}
let kii = {};
table3.forEach((t) => {
  Object.keys(t).forEach((y) => {
    kii[y] = 1;
  });
});
});

```

```

result.push(kii);
result.push(table3);

result.push(
  "Значення сингтонів частотності нормованих значень вагових коефіцієнтів"
);
f = [];
table3.forEach((el, i) => {
  f.push(
    Object.entries(el)
      .map(([k, v]) => (k && v ? `+μ${i + 1}({k})/${v}` : null))
      .filter((e) => e)
      .join("")
      .slice(1)
  );
});
result.push(f);

f = [];
let u = [];
result.push(
  "Здійснимо нормалізацію функцій належності і представимо нечіткі значення
відносних коефіцієнтів компетентності експертів у вигляді нормальних функцій
належності"
);

table3.forEach((el, i) => {
  const max = findMax(Object.values(el));
  u[i] = [];
  f.push(
    Object.entries(el)
      .map(([k, v]) => {
        u[i].push({ x: +k, y: Math.round((v / max) * 100) / 100 });
        return k && v
          ? `+μ${i + 1}({k})/${Math.round((v / max) * 100) / 100}`
          : null;
      })
      .filter((e) => e)
      .join("")
      .slice(1)
  );
});
result.push(u);
result.push(f);

result.push(
  "Частотність ідеалізованих значень вагових коефіцієнтів відносної
компетентності експертів у вибраних інтервалах"
);
let table4 = new Array(q.length);
const range2 = getRange(0.2);
for (let i = 0; i < table4.length; i++) {
  table4[i] = {};
  for (let u = 0; u < range2.length; u++) {
    table4[i][range2[u]] = 0;
  }
}
}

```

```

for (let i = 0; i < table2.length; i++) {
  for (let j = 0; j < table2[i].length; j++) {
    for (let k = 1; k < range2.length; k++) {
      if (table2[i][j] > range2[k - 1] && table2[i][j] <= range2[k]) {
        if (!table4[i][range2[k]]) {
          table4[i][range2[k]] = 1;
        } else {
          table4[i][range2[k]]++;
        }
      }
    }
  }
}
kii = {};
table4.forEach((t) => {
  Object.keys(t).forEach((y) => {
    kii[y] = 1;
  });
});
result.push(kii);
result.push(table4);
result.push(
  "Нечіткі нормовані вагові коефіцієнти компетентності експертів у вигляді
  синглтонів частотності значень вагових коефіцієнтів"
);
f = [];
table4.forEach((el, i) => {
  f.push(
    Object.entries(el)
      .sort(([k1,v1],[k2,v2]) => +k1-k2)
      .map(([k, v]) => (k && v ? `+μ${i + 1}(${k})/${v}` : null))
      .filter((e) => e)
      .join("")
      .slice(1)
  );
});
result.push(f);
f = [];
u = [];
result.push(
  "Синглтони для вираження нечітких значень вагових коефіцієнтів відносної
  компетентності експертів "
);
table4.forEach((el, i) => {
  const max = findMax(Object.values(el));
  u[i] = [];
  f.push(
    Object.entries(el)
      .sort(([k1,v1],[k2,v2]) => +k1-k2)
      .map(([k, v]) => {
        u[i].push({ x: +k, y: Math.round((v / max) * 100) / 100 });
        return k && v
          ? `+μ${i + 1}(${k})/${Math.round((v / max) * 100) / 100}`
          : null;
      })
      .filter((e) => e)
      .join("")
  );
});

```

```

        .slice(1)
      );
    });
    result.push(u);
    result.push(f);
    return result;
  };

```

Додаток 2

```

.App {
  text-align: center;
}

```

```

.App-logo {
  height: 40vmin;
  pointer-events: none;
}

```

```

@media (prefers-reduced-motion: no-preference) {
  .App-logo {
    animation: App-logo-spin infinite 20s linear;
  }
}

```

```

.App-header {
  background-color: #282c34;
  min-height: 100vh;
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
  font-size: calc(10px + 2vmin);
  color: white;
}

```

```

.App-link {
  color: #61dafb;
}

```

```

@keyframes App-logo-spin {
  from {
    transform: rotate(0deg);
  }
  to {
    transform: rotate(360deg);
  }
}

```

```
}

```

Додаток 3

```
import { useEffect, useState } from "react";
import "./App.css";
import InputData from "./components/InputData";
import InputMatrix from "./components/InputMatrix";
import { Table, Tr, Td, Thead, Th, Box, Tbody } from "@chakra-ui/react";
import Chart from "react-apexcharts";

import { start } from "./helpers/dataHelper";

function App() {
  const [inputSize, setInputSize] = useState();
  const [inputNumber, setInputNumber] = useState();
  const [inputMatrix, setInputMatrix] = useState([]);
  const [data, setData] = useState([]);

  return (
    <div>
      <InputData
        label="Кількість ранжувань"
        onSave={(e) => setInputNumber(e)}
      />
      <InputData label="Розмір ранжувань" onSave={(e) => setInputSize(e)} />
      {inputSize && inputNumber ? (
        <InputMatrix
          n={inputNumber}
          m={inputSize}
          onSave={(m) => {
            setInputMatrix(m);
            setData(start(m));
          }}
        />
      ) : null}
      {data.length ? (
        <>
          <Box>{data[0]}</Box>
          <br />
          <Box>{JSON.stringify(data[1])}</Box>
          <br />
          <Box>{data[2]}</Box>
          <br />
          <Box>{data[3]}</Box>
        </>
      ) : null}
    </div>
  );
}

```

```

<br />
<Box>{JSON.stringify(data[4])}</Box>
<br />
<Box>{data[5]}</Box>
<br />
<Box>{data[6]}</Box>
<br />
<Box>{JSON.stringify(data[7])}</Box>
<br />
<Box>{data[8]}</Box>
<br />
<Box>{data[9]}</Box>
<br />
<Box>{JSON.stringify(data[10])}</Box>
<br />
<Box>{data[11]}</Box>
<br />
<Box>{data[12]}</Box>
<br />
<Table>
  <Thead fontSize="45px">
    <Tr>
      <Th fontSize="45px">Ранжування</Th>
      <Th fontSize="45px">Частотність</Th>
      {data[13][0][2].map((_w, i) => (
        <Th fontSize="45px">{i + 1}</Th>
      ))}
    </Tr>
  </Thead>
  <Tbody>
    {data[13].map((e) => (
      <Tr>
        <Td>{e[0]}</Td>
        <Td>{e[1]}</Td>
        {e[2].map((r) => (
          <Td>{r}</Td>
        ))}
      </Tr>
    ))}
  </Tbody>
</Table>
<br />
<Box>{data[14]}</Box>

```

```

<br />
<Table>
  <Thead>
    <Tr>
      <Th fontSize="45px">Ранжування</Th>
      <Th fontSize="45px">Частотність</Th>
      {data[15][0][2].map((_w, i) => (
        <Th fontSize="45px">{i + 1}</Th>
      ))}
    </Tr>
  </Thead>
  <Tbody>
    {data[15].map((e) => (
      <Tr>
        <Td>{e[0]}</Td>
        <Td>{e[1]}</Td>
        {e[2].map((r) => (
          <Td>{r}</Td>
        ))}
      </Tr>
    ))}
  </Tbody>
</Table>
<br />
<Box>{data[16]}</Box>
<br />
<Table>
  <Thead>
    <Tr>
      <Th fontSize="45px">Ранжування</Th>
      <Th fontSize="45px">Частотність</Th>
      {data[17][0][2].map((_w, i) => (
        <Th fontSize="45px">{i + 1}</Th>
      ))}
    </Tr>
  </Thead>
  <Tbody>
    {data[17].map((e) => (
      <Tr>
        <Td>{e[0]}</Td>
        <Td>{e[1]}</Td>
        {e[2].map((r) => (
          <Td>{r}</Td>
        ))}
      </Tr>
    ))}
  </Tbody>
</Table>

```

```

        )))}
    </Tr>
    )))}
</Tbody>
</Table>
<br />
<Box>{data[18]}</Box>
<br />
<Table>
  <Thead>
    <Tr>
      <Th fontSize="45px">№</Th>
      {[...new Array(data[19].length)].map((_w, i) => (
        <Th fontSize="45px">{i + 1}</Th>
      ))}
    </Tr>
  </Thead>
  <Tbody>
    {data[19][0].map((e, t) => (
      <Tr>
        <Td>{t + 1}</Td>
        {[...new Array(data[19].length)].map((_r, i) => (
          <Td>{data[19][i][t]}</Td>
        ))}
      </Tr>
    ))}
  </Tbody>
</Table>
<br />
<Box>{data[20]}</Box>
<br />
<Table>
  <Thead>
    <Tr>
      <Th fontSize="45px">№</Th>
      {[...new Array(data[21].length)].map((_w, i) => (
        <Th fontSize="45px">{i + 1}</Th>
      ))}
    </Tr>
  </Thead>
  <Tbody>
    {data[21][0].map((e, t) => (
      <Tr>

```

```

    <Td>{t + 1}</Td>
    {[...new Array(data[21].length)].map((_r, i) => (
      <Td>{data[21][i][t]}</Td>
    ))}
  </Tr>
  )})
</Tbody>
</Table>
<br />
<Box>{data[22]}</Box>
<br />
<Table>
  <Thead>
    <Tr>
      <Th fontSize="45px">Значення</Th>
      {[...new Array(inputNumber)].map((_t, i) => (
        <Th fontSize="45px">{i + 1}</Th>
      ))}
    </Tr>
  </Thead>
  <Tbody>
    {Object.keys(data[23])
      .sort((a, b) => a - b)
      .map((el) => (
        <Tr>
          <Td>{el}</Td>
          {[...new Array(inputNumber)].map((_t, i) => (
            <Td>{data[24][i][el] || 0}</Td>
          ))}
        </Tr>
      ))}
  </Tbody>
</Table>
<br />
<Box>{data[25]}</Box>
<br />
<Box>
  {data[26].map((r) => (
    <span>
      {r}
    <br />
  </span>

```

```

    ))}
  </Box>
  <br />
  <Box>{data[27]}</Box>
  <br />
  <Box>
    {data[29].map((r, i) => {
      return (
        <span>
          {r}
          <br />
          <Chart
            series={[
              { name: i, data: data[28][i].sort((a, b) => a.x - b.x) },
            ]}
            width="500"
            type="line"
            options={{
              chart: {
                id: "basic-bar",
              },
              tooltip: {},
              grid: {
                row: {
                  colors: ["#f3f3f3", "transparent"], // takes an array which will be
repeated on columns
                  opacity: 0.5,
                },
              },
              xaxis: {
                type: "numeric",
                tickAmount: 20,
                decimalsInFloat: 2,
              },
            }}
          />
        </span>
      );
    })}
  </Box>
  <br />
  <Box>{data[30]}</Box>
  <br />

```

```

<Box>
  <Table>
    <Thead>
      <Tr>
        <Th fontSize="45px">Значення</Th>
        {[...new Array(inputNumber)].map((_t, i) => (
          <Th fontSize="45px">{i + 1}</Th>
        ))}
      <></>
    </Tr>
  </Thead>
  <Tbody>
    {Object.keys(data[31])
      .sort((a, b) => a - b)
      .map((el) => (
        <Tr>
          <Td>{el}</Td>
          {[...new Array(inputNumber)].map((_t, i) => (
            <Td>{data[32][i][el] || 0}</Td>
          ))}
        </Tr>
      ))}
    </Tbody>
  </Table>
</Box>
<br />
<Box>{data[33]}</Box>
<br />
<Box>
  {data[34].map((r) => (
    <span>
      {r}
    <br />
    </span>
  ))}
</Box>
<br />
<Box>{data[35]}</Box>
<br />
<Box>
  {data[37].map((r, i) => {
    return (
      <>

```

```

    <span>
      {r}
    <br />
  </span>
  <Chart
    series={[
      { name: i, data: data[36][i].sort((a, b) => a.x - b.x) },
    ]}
    width="500"
    type="line"
    options={{
      chart: {
        id: "basic-bar",
      },
      grid: {
        row: {
          colors: ["#f3f3f3", "transparent"], // takes an array which will be
repeated on columns
          opacity: 0.5,
        },
      },
      xaxis: {
        type: "numeric",
        tickAmount: 5,
        decimalsInFloat: 2,
      },
    }}
  />
</>
);
}}
</Box>
</>
): null}
</div>
);
}

```

```
export default App;
```

Додаток 4

```
import { render, screen } from '@testing-library/react';
import App from './App';
```

```
test('renders learn react link', () => {
  render(<App />);
  const linkElement = screen.getByText(/learn react/i);
  expect(linkElement).toBeInTheDocument();
});
```

Додаток 5

```
body {
  margin: 0;
  font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', 'Roboto', 'Oxygen',
    'Ubuntu', 'Cantarell', 'Fira Sans', 'Droid Sans', 'Helvetica Neue',
    sans-serif;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
  font-size: 45px;
}

code {
  font-family: source-code-pro, Menlo, Monaco, Consolas, 'Courier New',
    monospace;
}
```

Додаток 6

```
import React from "react";
import ReactDOM from "react-dom/client";
import "./index.css";
import App from "./App";
import { ChakraProvider } from "@chakra-ui/react";

const root = ReactDOM.createRoot(document.getElementById("root"));
root.render(
  <ChakraProvider>
    <App />
  </ChakraProvider>
);
```