МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ
Кафедра фундаментальної та прикладної математики

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему: **«ПОГАНО СТРУКТУРОВАНІ ПРОБЛЕМИ ТА ЇХ ДОСЛІДЖЕННЯ ЗАСОБАМИ ТЕОРІЇ ПРИЙНЯТТЯ РІШЕНЬ»**

Виконала: студентка 4 курсу, групи 6.1131

спеціальності 113 Прикладна математика
(шифр і назва спеціальності)

освітньої програми Комп'ютерне моделювання
(назва освітньої програми)

І. О. Лапенко
(ініціали та прізвище)

Керівник доцент кафедри фундаментальної та прикладної математики, доцент, к.ф.-м.н. Леонтьєва В. В.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Рецензент декан математичного факультету, д.т.н., професор Гоменюк С. І.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Запоріжжя – 2025

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет математичний

Кафедра кафедра фундаментальної та прикладної математики

Рівень вищої освіти бакалавр

Спеціальність 113 Прикладна математика

(шифр і назва)

Освітня програма Комп'ютерне моделювання

ЗАТВЕРДЖУЮ

Завідувач кафедрою фундаментальної та прикладної математики, д.т.н., професор

_____ Гребенюк С.М.

(підпис)

« ____ » _____ 2025 р.

З А В Д А Н Н Я

НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТЦІ

Лапенко Ілоні Олексіївні

(прізвище, ім'я та по-батькові)

1. Тема роботи (проекту) Погано структуровані проблеми та їх дослідження засобами теорії прийняття рішень

керівник роботи (проекту) Леонтьєва Вікторія Володимирівна, к.ф.-м.н., доцент

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

затверджені наказом ЗНУ від « 20 » 12 2024 року № 2001-с

2. Строк подання студентом роботи 02.06.2025 р.

3. Вихідні дані до роботи 1. Постановка задачі.
2. Перелік літератури.

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

1. Ключові концепти теорії прийняття рішень основні відомості стосовно погано структурованих проблем
2. Огляд основних методів теорії прийняття рішень та їх застосування до розв'язання погано структурованих проблем
3. Огляд задач для реалізації у програмному середовищі
4. Автоматизація дослідження та розв'язання практичних погано структурованих проблем

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

Презентація

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 20.12.2024

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Розробка плану роботи.	10.01.2025-22.02.2025	
2.	Збір вихідних даних.	6.03.2025-12.03.2025	
3.	Обробка методичних та теоретичних	16.03.2025-26.03.2025	
	джерел.		
4.	Розробка першого розділу.	2.04.2025-10.04.2025	
5.	Розробка другого розділу.	14.04.2025-26.04.2025	
6.	Розробка третього розділу.	29.04.2025-4.05.2025	
7.	Розробка четвертого розділу.	7.05.2025-14.05.2025	
8.	Створення програмних продуктів та їх	15.05.2025-30.05.2025	
	тестування.		
9.	Оформлення та нормоконтроль	1.06.2025-16.06.2025	
	кваліфікаційної роботи.		
10.	Захист кваліфікаційної роботи.	17.06.2025	

Студент _____ І. О. Лапенко
(підпис) (ініціали та прізвище)

Керівник роботи _____ В. В. Леонтьєва
(підпис) (ініціали та прізвище)

Нормоконтроль пройдено

Нормоконтролер _____ О. Г. Спиця
(підпис) (ініціали та прізвище)

РЕФЕРАТ

Кваліфікаційна робота бакалавра «Погано структуровані проблеми та їх дослідження засобами теорії прийняття рішень»: 51 с., 10 рис., 12 джерел.

АВТОМАТИЗАЦІЯ, КРИТЕРІЇ ПРИЙНЯТТЯ РІШЕНЬ, ПОГАНО СТРУКТУРОВАНІ ПРОБЛЕМИ, ПРОБЛЕМА, ПРОГРАМНИЙ ПРОДУКТ, ТЕОРІЯ ПРИЙНЯТТЯ РІШЕНЬ.

Об’єкт дослідження – погано структуровані проблеми.

Предмет дослідження – особливості погано структурованих проблем, підходи і методи до їх розв’язання засобами теорії прийняття рішень.

Мета роботи: аналіз погано структурованих проблем методами теорії прийняття рішень; розроблення програмного забезпечення для автоматизації процесу розв’язання погано структурованих проблем.

Методи дослідження: аналітичний, порівняльний, практичний.

Структурно робота поділяється на чотири розділи. У першому розділі наводяться різноманітні теоретичні відомості з теорії прийняття рішень, представляється класифікація проблем, а особлива увага приділяється погано структурованим проблемам, надається постановка задачі прийняття рішень в умовах невизначеності та ризику, розглядаються розподіли задач прийняття рішень за певними ознаками. У другому розділі аналізуються особливості методів теорії прийняття рішень та їх використання при розв’язанні погано структурованих проблем, описуються та пояснюються критерії теорії прийняття рішень. Третій розділ присвячений опису задач, розв’язок котрих буде реалізований у наступному розділі шляхом написання відповідних програм. У четвертому розділі наводяться програмні реалізації практичних погано структурованих проблем за допомогою критеріїв прийняття рішень. За кожною з розроблених програм проводяться тестування за трьома різними вхідними даними для кожної задачі, висвітлюються покрокові алгоритми роботи програм, які є інструкціями для кінцевого користувача.

SUMMARY

Bachelor’s Qualifying Theses "Ill-Structured Problems and Their Research Using Methodology of Decision-Making Theory": 51 pages, 10 figures, 12 references.

AUTOMATION, DECISION-MAKING CRITERIA, ILL-STRUCTURED PROBLEMS, PROBLEM, SOFTWARE PRODUCT, DECISION-MAKING THEORY.

Object of the study is ill-structured problems.

The subject of research is the features of ill-structured problems, approaches and methods for their solution using decision theory.

Aim of the study is analysis of ill-structured problems using decision theory methods; development of software to automate the process of solving ill-structured problems.

Methods of research are analytical, comparative, practical.

The paper is organized into four sections. The first section provides various theoretical information on decision theory, presents a classification of problems, with special attention to poorly structured problems, provides a statement of the decision-making problem under conditions of uncertainty and risk, and considers the distribution of decision-making problems by certain characteristics. The second section analyzes the features of decision theory methods and their use in solving poorly structured problems, describes and explains the criteria of decision theory. The third section is devoted to the description of tasks, the solution of which will be realized in the next section by writing the corresponding programs. The fourth section presents program implementations of practical ill-structured problems using decision-making criteria. Each of the developed programs is tested with three different input data for each problem, and step-by-step algorithms of the programs are highlighted, which are instructions for the end user.

ЗМІСТ

Завдання на кваліфікаційну роботу……..……..……... 2

Реферат……. 4

Summary.……...…… 5

Вступ……. 8

1 Ключові концепти теорії прийняття рішень та основні відомості стосовно погано структурованих проблем……..… 10

1.1 Теорія прийняття рішень: історія, розвиток та основні положення……..…… 10

1.2 Постановка задачі прийняття рішень…… 12

1.3 Види задач теорії прийняття рішень……. 13

1.4 Класифікація рішень та процес їх прийняття…… 14

1.5 Проблеми та їх розподіл за ступенем структурованості……. 17

1.6 Погано структуровані проблеми та їх властивості…… 18

2 Огляд основних методів теорії прийняття рішень та їх застосування до розв’язання погано структурованих проблем 20

2.1 Основні методи теорії прийняття рішень…….. 20

2.2 Застосування критеріїв теорії прийняття рішень при розв’язанні погано структурованих проблем……..... 22

3 Огляд задач для реалізації у програмному середовищі …… 28

3.1 Задача про вибір найбільш доцільної стратегії для підприємства у період кризових економічних умов……..... 28

3.2 Задача про вибір оптимального місця розташування для будівництва укриття або пункту незламності…….……... 29

3.3 Задача про планування маршрутів для безпечного проїзду вантажівок…… 30

4 Автоматизація дослідження та розв’язання практичних погано структурованих проблем……... 31

4.1 Засоби реалізації розроблюваних програмних продуктів…….. 31
4.2 Програмна реалізація задачі про вибір найбільш доцільної стратегії для підприємства у період призових економічних умов……... 32
4.3 Програмна реалізація задачі про вибір оптимального місця для будівництва укриття або пункту незламності…… 37
4.4 Програмна реалізація задачі про планування маршрутів для безпечного проїзду вантажівок…… 42
Висновки……. 48
Перелік посилань……..…….....…….. 50

ВСТУП

Кожного дня приймається величезна кількість рішень. Це відбувається у різних галузях науки, освіти, спорту, інформатики, економіки, математики, інженерії, екології, політики, філології, культури та у багатьох інших сферах [1]. Ухвалення рішень може відбуватися за різних обставин. Дуже часто їх ухвалюють в умовах відсутності достатньої інформації, високої складності, невизначеності, обмеженості інструментів та ресурсів. Такі умови свідчать про погано структуровані проблеми, які і досліджує теорія прийняття рішень. Саме такі проблеми розглядаються, аналізуються та розв'язуться у даному дослідженні [12].

Актуальність цього дослідження зумовлена необхідністю аналізу проблем та задач, які характеризуються підвищеною складністю, неповнотою наданої інформації та її суперечливістю. За таких обставин стандартні методи можуть бути недостатніми, малоефективними для розв'язання, адже такі задачі можуть потребувати особливих підходів, які і засновуються на теорії прийняття рішень. Погано структуровані задачі характеризуються обчислювальною складністю, тому виникає необхідність їх програмної реалізації.

Мета роботи полягає в аналізі погано структурованих проблем методами теорії прийняття рішень та розробленні програмного забезпечення для автоматизації процесу розв'язання погано структурованих проблем.

Об'єктом дослідження виступають погано структуровані проблеми.

Структурно робота поділяється на чотири розділи.

У першому розділі наводяться різноманітні теоретичні відомості з теорії прийняття рішень, представляється класифікація проблем, а особлива увага приділяється погано структурованим проблемам, надається постановка задачі прийняття рішень в умовах невизначеності та ризику, розглядаються розподіли задач прийняття рішень за певними ознаками.

У другому розділі аналізуються особливості методів теорії прийняття рішень та їх використання при розв’язанні погано структурованих проблем, описуються та пояснюються критерії теорії прийняття рішень.

Третій розділ присвячений опису задач, розв’язок котрих буде реалізований у наступному розділі шляхом написання відповідних програм.

У четвертому розділі наводяться програмні реалізації практичних погано структурованих проблем, що були сформульовані та описані у попередньому розділі, за допомогою критеріїв Гурвиця, Вальда та Байєса-Лапласа. За кожною з розроблених програм проводяться тестування за трьома різними вхідними даними для кожної задачі. Разом із цим висвітлюються покрокові алгоритми роботи програм, які фактично є інструкціями для кінцевого користувача.

1 КЛЮЧОВІ КОНЦЕПТИ ТЕОРІЇ ПРИЙНЯТТЯ РІШЕНЬ ТА ОСНОВНІ ВІДОМОСТІ СТОСОВНО ПОГАНО СТРУКТУРОВАНИХ ПРОБЛЕМ

Даний розділ – присвячений огляду основних концепцій, історичних аспектів, ключових особливостей теорії прийняття рішень, а також класифікаціям задач теорії прийняття рішень за ознаками, перелікам рішень та етапам їх ухвалення. У ньому наводиться розподіл проблем за ступенем їх структурованості, висвітлюються особливості погано структурованих проблем. Разом із цим описується оформлення важливого етапу у прийнятті рішення – постановки задачі.

1.1 Теорія прийняття рішень: історія, розвиток та основні положення

Перш за все, в даному підрозділі визначимось з ключовими відомостями теорії прийняття рішень. Розглянемо її формування, як наукової дисципліни.

Необхідність приймати рішення завжди було важливим аспектом у житті людей. Ще до формування теорії прийняття рішень, люди вирішували її різні задачі. А завдяки науковому прогресу теорія прийняття рішень, як наукова галузь почала формуватися з середини минулого сторіччя. Також утворенню даної сфери наукової діяльності сприяв стрімкий розвиток інформаційних технологій, програмування, інформатики та інших галузей людської діяльності. Зараз же теорія прийняття рішень – це невід'ємна наука, яка допомогає розв'язувати задачі різних видів, складностей та умов [6].

Далі розглянемо визначення теорії прийняття рішень та її особливості.

Теорія прийняття рішень – це така область наукової діяльності, яка аналізує різні альтернативи та допомагає зробити вибір, що спрямований на

найбільш оптимальний варіант за обставин визначеності, невизначеності чи ризику [3]. Зазвичай кажуть, що це міждисциплінарна наукова галузь. Теорія прийняття рішень пов'язана з багатьма сферами людської діяльності: наукою, освітою, спортом, психологією, військовою сферою, статистикою, менеджментом, інформатикою, економікою, математикою, інженерією, екологією, політикою, філологією, культурою та з іншими сферами [1].

Провідні особливості теорії прийняття рішень полягають у наступному:

- широке застосування – як зазначено вище, теорія прийняття рішень є міждисциплінарною областю наукових знань, це означає, що її методологічні засоби тісно пов'язані з різними предметами людської діяльності. Наприклад: з психологією, математикою, інформатикою, статистикою та з іншими галузями;
- наявність альтернативних рішень – означає, що при розв'занні задач теорія прийняття рішень дозволяє обрати найбільш доцільний та найбільш ефективний підхід з кількох можливих;
- можливість рішення задач різної складності та за різних умов. Наприклад: за умов невизначеності, визначеності та ризику;
- пріоритним є найбільш доцільне рішення серед інших – означає, що вибір теорії прийняття рішень падає на найбільш збілансований варіант серед існуючих.

Визначимося з об'єктом та предметом дослідження теорії прийняття рішень, її ідеєю.

Об'єктом дослідження виступає проблемна ситуація, тобто така ситуація, яка потребує обрати серед деякої кількості альтернатив оптимальну [2].

Предмет дослідження – це певні правила, які застосовують для ухвалення оптимальних рішень, вибору найдоцільнішого варіанту з усіх [2].

Ідея теорії прийняття рішень полягає у побудові таких алгоритмів, які допожуть обрати найбільш збалансоване рішення серед наявних. При цьому

алгоритми враховують наявні обмеження, складність задачі, кількість вхідних даних тощо [3].

Як зазначено вище, теорія прийняття рішень бере свій початок ще з середини XX століття. Відомо, що люди активно користувалися її методами та вона стрімко розвивалася. Наразі теорія прийняття рішень є важливою науковою дисципліною. Потреба у її використання полягає у великій кількості проблем, що мають високий рівень складності, недостатню кількість вхідних даних та є погано структурованими. Саме такі складні проблеми теорія прийняття рішень здатна розв'язувати, аналізувати, формалізувати, розглядати, оцінювати тощо. У цій роботі дані проблеми виступають об'єктом дослідження, тому вони обов'язково будуть розглядатися.

1.2 Постановка задачі прийняття рішень

Постановка задачі прийняття рішень – це важливий етап у вирішенні будь-якої проблеми теорії прийняття рішень [8].

Для того, щоб коректно оформити постановку задачі прийняття рішення необхідно з'ясувати:

- яку суть має проблема, котру треба вирішити та за яких умов ця проблема виникла;
- за який термін потрібно прийняти рішення;
- за допомогою чого буде розв'язана проблема, за яких ресурсів.

З метою аналізу проблеми, визначення обмежень, структуризації суперечливих аспектів та визначення складностей необхідно виконати наступне:

- визначити проблематику завдання та надати його характеристику;
- скільки часу потрібно на розв'язання завдання;
- проаналізувати доступні ресурси та інструменти.

Як було зазначено вище, формулювання задачі прийняття рішення – це значущий етап, необхідний для характеристики та опису поставленої задачі, визначення її проблемних аспектів, аналізу наявних ресурсів та кількості часу, а також для досягнення успішного результату.

Розглянемо формальне представлення постановки задачі теорії прийняття рішень. Вона має наступний вигляд:

$$A = (B, B^E, D, I^E, F^E, G, E), \tag{1.1}$$

де B – набір імовірних ситуацій, B^E – набір проблемних ситуацій, I^E – набір постановок задач теорії прийняття рішень, G – набір імовірних розв'язків, E – набір пріоритетів особи, що приймає рішення.

1.3 Види задач теорії прийняття рішень

Теорія прийняття рішень розглядає велику кількість різних задач. Для того, щоб краще зрозуміти, чим вони відрізняються, розглянемо декілька класифікацій таких задач [4, 7].

Розподіл задач за кількістю критеріїв при їх постановці:

- з одним критерієм – це задачі, при постановці яких враховують тільки один критерій;
- з багатьма критеріями – це такі задачі, при постановці яких враховують велику кількість критеріїв.

Розподіл задач за швидкістю ухвалення рішень:

- динамічні – такий тип задач, в яких рішення ухвалюють поступово;
- статичні – такі задачі, в котрих рішення ухвалюють за один раз.

Види задач за умовами, в яких вони знаходяться:

- умови ризику – такі умови, при яких відома думка експертів або статистика про можливі наслідки;

- умови визначеності – такі умови, за яких всі параметри відомі та незмінні з плином часу;
- умови часткової невизначеності – це умови, при яких інформація про параметри чи наслідки надана частково, чи наближено;
- умови абсолютної невизначеності – це умови, за яких інформація про параметри чи наслідки повністю відсутня;
- умови конфлікту – це такі умови, котрі обумовлюються присутністю деяких інших осіб, думки яких протирічать думкам суб'єкта прийняття рішень.

Як відомо, теорія прийняття рішень розглядає погано структуровані проблеми за умов невизначеності та ризику [5]. При таких обставинах достатньо складно прийняти правильне рішення, передбачити результат подальших дій з високою точністю, чітко обрати потрібні алгоритми. Тому існують різноманітні критерії, наприклад, Гурвиця, Вальда, Байєса-Лапласа. Більш детально вони будуть розглядатися у наступному розділі.

1.4 Класифікація рішень та процес їх прийняття

У даному підрозділі приділимо увагу таким поняттям як рішення та процес прийняття рішень, адже це невід'ємна складова теорії прийняття рішень.

Перш за все оглянемо поняття рішення.

Рішення – деякий вибір людини, котрий веде до певних наслідків [1]. Також рішенням називають обрання найкращої альтернативи, серед можливих альтернатив в рамках деякої ситуації з метою досягнення необхідного результату.

Далі оглянемо значення процесу прийняття рішення.

Процес прийняття рішення – це набір послідовних дій, котрі ведуть до обрання певного рішення.

Для кращого розуміння підрозділу, розглянемо етапи покрокового прийняття рішень:

- аналіз наявної ситуації;
- оформлення постановки задачі;
- утворення мети, розв’язок проблеми;
- з’ясування, знаходження різних варіантів досягнення поставленої мети;
- аналіз зовнішнього середовища;
- знаходження імовірних наслідків від дій;
- розгляд і оцінка очікуваних наслідків виконання варіантів за певних умов зовнішнього середовища;
- визначитися чи відповідають отримані результати запланованим;
- аналіз запланованих наслідків;
- вибір найоптимальнішого варіанту серед наявних шляхом порівняння;
- остаточний вибір оптимального варіанту, розв’язання проблемної ситуації, затвердження алгоритму дій.

Далі розглянемо різноманітні види рішень [1, 10].

Розподіл рішень за масштабом:

- глобальні – важливі рішення, що чинять масштабний вплив на щось;
- локальні – це такі рішення, котрі не виходять за певні рамки.

Види рішень за першоджерелом їх утворення:

- по розпорядженню – рішення, які приймає вище керівництво по розпорядженню, за наказом;
- ініціативні – рішення, котрі приймають за бажанням особи, тобто за її ініціативою;
- за замовленням – такі види рішень, котрі були ухвалені за бажанням замовника.

За способом викладу інформації рішення подяляють на:

- усні – такі рішення, які викладають в усній формі;
- письмові – рішення, які викладають у письмовій формі.

Розподіл за кількістю осіб, котрі ухвалюють рішення:

- колегіальні – рішення, що ухвалюють кілька осіб;
- індивідуальні – рішення, що приймає одна особа;
- колективні – рішення, котрі приймає весь колектив.

В залежності від кількості відомої інформації, рішення поділяють на:

- визначені – рішення, котрі приймають, володіючи необхідною інформацією;
- ймовірні – рішення, які ухвалюють володіючи неповною кількістю;
- невизначені – рішення, котрі приймають, не володіючи необхідною інформацією або вона взагалі відсутня.

Розподіл рішень за кількістю цілей:

- з однією – рішення, котрі мають лише одну ціль;
- з багатьма – рішення, котрі мають багато цілей.

В залежності від їх змісту, рішення бувають наступні:

- економічні – ті, що пов’язані з економікою, фінансами, витратами тощо;
- технічні – ті, що пов’язані з інженерією тощо;
- соціальні – ті, що пов’язані з соціумом, населенням тощо;
- організаційні – ті, що пов’язані з організацією, управлінням тощо.

За часом, швидкістю прийняття рішення поділяються на:

- довготривалі – рішення, для прийняття яких потрібен довгий проміжок часу;
- оперативні – рішення, для ухвалення яких потрібен короткий проміжок часу.

Розрізняють за рівнем усвідомленості:

- свідомі рішення – рішення, що прийняті після повного аналізу ситуації;
- погано усвідомлені – рішення, що прийняті після часткового аналізу ситуації;
- несвідомі рішення – рішення, що прийняті без аналізу ситуації.

Отже, в рамках даного підрозділу було розглянуто рішення, покроковий процес його прийняття та розподіл рішень за ознаками.

1.5 Проблеми та їх розподіл за ступенем структурованості

Проблема – це непроста ситуація чи певної складності питання, котре треба вирішити, опрацювати та дослідити [10].

В залежності від визначеності цілей, наявності чітких алгоритмів для розв'язання та кількості відомої інформації проблеми розподіляють на наступні категорії: добре структуровані, слабоструктуровані(або погано структуровані), неструктуровані.

Розглянемо види проблем, за рівнем їх структурованості [7, 8]:

- добре структуровані проблеми – це такі типи проблем, які характеризуються чітко визначеними цілями, мають повну інформацію та швидко розв'язуються звичними методами і алгоритмами. Кажуть, що це кількісно описані проблеми;
- слабо структуровані проблеми або погано структуровані проблеми – це проблеми, які характеризуються відсутністю чітких цілей, високим рівнем невизначеності та відсутністю чітких алгоритмів вирішення. Зазвичай кажуть, що це комбіновані проблеми;
- неструктуровані проблеми – це такі проблеми, які характеризуються відсутністю цілей, складністю та неможливістю розв'язання

звичайними алгоритмати, методами. Кажуть, що це якісно описані проблеми.

Розуміння ступеня структуризації проблеми дає змогу підібрати відповідні методи та засоби теорії прийняття рішень.

1.6 Погано структуровані проблеми та їх властивості

Оскільки об’єктом дослідження даної роботи виступають погано структуровані проблеми, то у даному підрозділі висвітлюються особливості проблем з поганою структуризацією, наводяться їхні властивості та види. Описуються такі проблеми у реальному світі.

Перш за все визначимось з поняттям погано структурованої проблеми.

Погано структуровані проблеми – це такі типи проблем, які характеризуються відсутністю чітко визначених цілей, високим ступенем невизначеності та відсутністю чітких алгоритмів для їх розв’язання [11].

Для кращого розуміння такого типу проблем нижче розглянемо їх властивості:

- високий рівень невизначеності – означає, що надані умови задачі можуть бути неповними, тобто надані частково;
- нечітко визначені цілі – цілі можуть бути багатозначними, змінюватися або мати конфліктний характер;
- надана інформація може бути неповною, суперечливою або нечіткою
- відсутня необхідна кількість вхідної інформації ;
- наявність певних ризиків – свідчить про те, що розв’язання проблем такого рівня складності може супроводжуватися виникненням помилок, великими затратами часу та ресурсів.

Далі розглянемо найпоширеніші види погано структурованих проблем:

- проблеми, котрі не потребують негайного вирішення;

- проблеми, котрим притаманний величезний велику кількість методів їх рішення;
- проблеми, для розв’язання яких необхідно мати достатньо ресурсів та розраховувати на можливу наявність різного роду ризиків;
- проблеми, для яких невідомо, яка кількість часу та грошей потрібні для їх вирішення [5].

Як було зазначено раніше, проблеми підвищеної складності виникають у найрізноманітніших сферах. Вони можуть викликати труднощі під час їх розв’язання. Тому необхідно розуміти особливості таких проблем, знати їх види, властивості та вміти їх визначати. Це дасть змогу підбирати правильні методи для вирішення проблем такаго типу, адже використання некоректних методів розв’язання може бути не тільки неефективним, а й може супроводжуватися великими затратами часу та ресурсів.

У контексті дослідження погано структурованих проблем, теорія прийняття рішень надає методологічну базу для опису невизначених параметрів, суперечливих цілей і нечітких критеріїв, що дозволяє перетворити складну задачу в керовану [2].

Отже, у поточному розділі було досліджено основні відомості теорії прийняття рішень, надано інформацію про її особливості та деякі історичні факти. Було надано класифікації проблем за рівнем їх структурованості, а також розподіли задач та рішень за різними ознаками. Розглянуто етапи прийняття рішень, процес постановки задачі та її представлення у формальному вигляді.

Даний розділ дозволяє здобути розуміння про об’єкт дослідження – погано структуровані проблеми. Він є основою для подальшого дослідження.

2 ОГЛЯД ОСНОВНИХ МЕТОДІВ ТЕОРІЇ ПРИЙНЯТТЯ РІШЕНЬ ТА ЇХ ЗАСТОСУВАННЯ ДО РОЗВ'ЯЗАННЯ ПОГАНО СТРУКТУРОВАНИХ ПРОБЛЕМ

Основна увага у цьому розділі приділяється методам теорії прийняття рішень, а особливо – критеріям, які використовують у теорії статистичних рішень. Також тут наводяться класифікації методів за ознаками, описи критеріїв та висновки.

2.1 Основні методи теорії прийняття рішень

Об'єктом дослідження даного підрозділу виступають методи теорії прийняття рішень. Також у ньому розглядаються розподіли таких методів за різноманітними ознаками. Наводяться визначення.

Під методом будемо розуміти спосіб досягнення поставленої мети, рішення задачі, шляхом застосування певних інструментів та засобів.

Оглянемо методи теорії прийняття рішень, які розподілятимуться за різними параметрами [9].

Так, за кількістю осіб, що приймають рішення, методи поділяють на:

- індивідуальні – коли рішення ухвалює одна особа самостійно або залучає експертів;
- групові – коли всі особи, що ухвалюють рішення в певній мірі мають декотрі обов'язки за наслідки вибору.

Різновиди методів за кількістю критеріїв виділяють наступні:

- однокритеріальні – тобто такі методи, провідна ідея яких лежить у концепціях пошуку найбільш оптимальної альтернативи за одним критерієм;

– багатокритеріальні – тобто такі методи, провідна ідея яких лежить у концепціях пошуку найбільш оптимальної альтернативи за кількома критеріями.

За використанням експертної підтримки методи поділяються на такі:

– із використанням експертної думки – мається на увазі, що в обчисленнях та обробці даних братимуть участь експерти;
– без використання експертної думки – означає, що в обчисленнях та обробці даних експерти не братимуть участі.

Види методів за типом аналізу інформації поділяють на:

– кількісні – це такі види методів, основна ідея яких полягає у чіткості, вибору найоптимальнішого варіанту, числових даних, математична інтерпритація результатів;
– якісні – це такі види методів, основна ідея яких полягає в залученні експертів, суб'єктивній інформації, що застосовується при виникненні труднощів з математичною інтерпретацією результатів;
– змішані – це такі види методів, які поєднують у собі ідеї як кількісних, так і якісних методів.

Розподіл методів за рівнем комп'ютеризації поділяють на:

– комп'ютеризовані – це такі типи методів, котрі обумовлені використанням різноманітної комп'ютерної обробки даних та інформації;
– ручні – до цієї категорії належать такі методи, котрі не передбачають використання комп'ютерів для обчислень;
– комбіновані – це такі методи, які передбачають використання як людської праці, так і комп'ютерних систем.

Основним методом розв'язання погано структурованих проблем прийнято вважати методи теорії статистичних рішень, заснованих на виборі того чи іншого критрію теорії прийняття рішень. Зазвичай, в широкому розумінні, під критеріями розуміють правила, визначальним особливостями

яких є можливість вибору найбільш вдалого варіанту з наявних, з врахуванням певних параметрів.

Більше про критерії та їх застосування в рамках теорії прийняття рішень розглянемо у наступному підрозділі.

2.2 Застосування критеріїв теорії прийняття рішень при розв'язанні погано структурованих проблем

В умовах невизначеності та ризику важко прийняти правильне рішення, передбачити результат подальших дій з високою точністю, чітко обрати потрібні алгоритми. Саме за таких обставин найчастіше і зустрічаються погано структуровані проблеми. Як було зазначено раніше, таким проблемам та задачам притаманні високий рівень невизначеності, підвищена складність, відсутність необхідної кількості вхідної інформації тощо. Часто, за таких характеристик, стандартні методи можуть бути безрезультатними або малоефективними. Але, щоб так не було, доцільно користуватися певними інструментами та засобами теорії прийняття рішень. Адже теорія прийняття рішень передбачає вивчення слабо структурованих проблем за обставин невизначеностей та ризиків [5]. Зважаючи на це, у даному підрозділі проаналізуємо поняття критерію та розглянемо найбільш поширені його різновиди, за допомогою яких розв'язують проблеми вище зазначеного типу.

З'ясуємо поняття критерію в теорії прийняття рішень.

Критерій – це так званий еталон порівняння, котрий слугує як основа для порівняння наявних рішень та вибору найбільш вдалого та доцільного серед інших.

Зважаючи на те, що теорія прийняття рішень вивчає погано структуровані проблеми за умов невизначеності та ризику, розглянемо критерії, які підходять для розв'язання проблем такого типу.

Оглядатимемо такі критеріїї: Гермейера, Ходжа-Лемана, Вальда, Байєса-Лапласа, Гурвиця, Севіджа.

Критерій Гермейєра – це такий максимінний критей, який містить у собі як оптимістичні, так і песимістичні стратегії. Дає змогу робити зважені рішення в умовах присутності неповної визначеності.

Критерій Гермейєра реалізується наступним чином:

$$\mathrm{G} = \max_{\mathrm{i}} \mathrm{e}_{\mathrm{ir}}; \tag{2.1}$$

$$\mathrm{e}_{\mathrm{ir}} = \min_{\mathrm{j}} \mathrm{e}_{\mathrm{ij}}\mathrm{q}_{\mathrm{j}}. \tag{2.2}$$

Оптимальна альтернатива обирається з наступного співвідношення:

$$\mathrm{E}_0 = \left\{\mathrm{E}_{\mathrm{i0}} | \mathrm{E}_{\mathrm{i0}} \in \mathrm{E} \wedge \mathrm{e}_{\mathrm{i0}} = \max_{\mathrm{i}} \min_{\mathrm{j}} \mathrm{e}_{\mathrm{ij}}\mathrm{q}_{\mathrm{j}} \wedge \mathrm{e}_{\mathrm{ij}} < 0\right\}. \tag{2.3}$$

Оглянемо правило вибору рішення за критерієм.

До матриці розв'язків $||e_{ij}||$ дописують стовпець, котрий має у кожному рядку найменший добуток її результату на ймовірність відповідного стану F_{j} .

Обирають такі альтернативи E_{i0}, у рядках котрих наявне найбільше e_{ir} даного стовпця.

Критерій Ходжа-Лемана – це такий показник, котрий передбачає поєднання критерів Баєса – Лапласа та максимінного критерію, Спрямований на обранні такої дії, котра б відповідала як середньому значенню, так і враховувала б найгірше значення.

Обчислення за критерієм проводиться наступному вигляді:

$$C_{HL} = \max_{\mathrm{i}} \mathrm{e}_{\mathrm{ir}}; \tag{2.4}$$

$$e_{ir} = v \sum_{j=1}^{n} e_{ij}\, \mathrm{q}_{\mathrm{j}} + (1 - \mathrm{v}) \min_{\mathrm{j}} e_{ij}\,,\ 0 \le v \le 1. \tag{2.5}$$

Оптимальну альтернативу обирають з наступного співвідношення:

$$\mathrm{E_0} = \left\{\mathrm{E_{i0}} | \mathrm{E_{i0}} \in \mathrm{E} \wedge \mathrm{e_{i0}} = \max_{\mathrm{i}} \left[\mathrm{v} \sum_{j=1}^{n} e_{ij}\, \mathrm{q_j} (1-\mathrm{v}) \min_{\mathrm{j}} \mathrm{e_{ij}}\right] \wedge 0 \leq \mathrm{v} \leq 1\right\}. \quad (2.6)$$

Правило вибору рішення за критерієм Ходжа-Лемана:

До матриці рішень $||e_{ij}||$ додають стовпець, котрий поєднує у собі середні зважені (з незмінними вагами) математичного сподівання й найбільш малого результату кожного рядка (2.5). Вибирають такі альтернативи рішень $\mathrm{E_0}$, у чиїх рядках знаходиться найбільше значення даного стовпця.

Критерій Вальда – зазвичай кажуть, що це максимінний критерій, котрий забезпечує максимізацію мінімального виграшу чи навпаки., заснований на крайньому песимізмі та на гарантованому результаті.

Розглянемо бчислення за критеієм Вальда:

$$W = \max_{\mathrm{i}} \mathrm{e_{ir}}; \quad (2.7)$$

$$\mathrm{e_{ir}} = \min_{\mathrm{j}} \mathrm{e_{ij}}. \quad (2.8)$$

Оптимальне рішення обирають з такого співвідношення:

$$\mathrm{E_0} = \left\{\mathrm{E_{i0}} | \mathrm{E_{i0}} \in \mathrm{E} \wedge \mathrm{e_{i0}} = \max_{\mathrm{i}} \min_{\mathrm{j}} \mathrm{e_{ij}}\right\}, \quad (2.9)$$

Розглянемо правило обрання рішення.

До матриці рішень $\mathrm{e_{ij}}$ додають додатковий стовпець, у якому для кожного рядка визначається мінімальне значення e_{ir}, вибирають альтернативи $\mathrm{E_{i0}}$, які відповідають рядкам із найбільшими серед знайдених мінімумів e_{ir} у новому стовпці.

Критерій Байєса-Лапласа – критерій, що заснований на виборі такої дії, яка б відповідала максимально очікуваному виграшу за наявності відомих або рівноймовірних станів середовища.

Нехай q_j – ймовірність, того, що з'явиться зовнішнішній стан F_j, тоді:

$$C_{BL} = \max_i e_{ir}; \quad (2.10)$$

$$e_{ir} = \sum_{j=1}^{n} e_{ij}\, q_j; \quad (2.11)$$

$$E_0 = \left\{E_{i0} | E_{i0} \in E \wedge e_{i0} = \max_i \sum_{j=1}^{n} e_{ij}\, q_j \wedge \sum_{j=1}^{n} q_j = 1\right\}. \quad (2.12)$$

Правило вибору рішення, відповідно до критерія Баєса-Лапласа: у матрицю рішень $||e_{ij}||$ додається додатковий стовпець, в якому для кожного рядка обчислюється математичне сподівання його елементів. Перевагу надають тим альтернативам E_{i0}, де відповідне значення математичного очікування e_{ir} є найбільшим серед усіх рядків.

Критерій Гурвиця – в основі баланс між найкращим та найгірщим результатом, в залежності від обраного рівня упевненості особи що прийм рішення.

Обчислення критерію Гурвиця:

$$H = \max_i e_{ir}; \quad (2.13)$$

$$e_{ir} = \alpha \min_j e_{ij} + (1 - \alpha) \max_j e_{ij}. \quad (2.14)$$

Оглянемо альтернативи:

$$E_0 = \left\{E_{i0} | E_{i0} \in E \wedge e_{i0} = \max_i \left[\alpha \min_j e_{ij} + (1 - \alpha) \max_j e_{ij}\right] \wedge 0 \leq \alpha \leq 1\right\}, \quad (2.15)$$

де α – ваговий множник.

Правило вибору за критерієм.

До матриці рішень $||e_{ij}||$ дописують стовпець, котрий включає середні зважені найбільш малого та найбільш великого результатів для кожного рядка (2.14). Вибирають такі альтернативи $\mathrm{E_{i0}}$, в чиїх рядках знаходяться найбільш великі елементи $\mathrm{e_{ir}}$ даного стовпця.

Критерій Севіджа – це такий показник, в основі якого лежить мінімізування максимальної кількості жалю.

Розглянемо реалізацію критерію Севіджа:

$$a_{ij} = \max_{\mathrm{i}} e_{ij} - e_{ij}; \tag{2.16}$$

$$\mathrm{e_{ir}} = \max_{\mathrm{j}} \mathrm{a_{ij}} = \max_{\mathrm{j}} (\max_{\mathrm{i}} e_{ij} - e_{ij}). \tag{2.17}$$

Функція оцінки за критерієм Севіджа:

$$S = \min_{\mathrm{i}} \mathrm{e_{ir}} = \min_{\mathrm{i}} \left[\max_{\mathrm{j}} \left(\max_{\mathrm{i}} e_{ij} - e_{ij}\right)\right]. \tag{2.18}$$

Множина альтернатив:

$$\mathrm{E_0} = \left\{\mathrm{E_{i0}} | \mathrm{E_{i0}} \in \mathrm{E} \wedge \mathrm{e_{i0}} = \min_{\mathrm{i}} \mathrm{e_{ir}}\right\}. \tag{2.19}$$

Нижче наводиться правило, за яким обирається відповідне рішення, згідно з критерієм Севіджа.

Кожний елемент матриці розв’язків $||e_{ij}||$ мінусують від найбільш великого результату $\max_{\mathrm{i}} e_{ij}$ відповідного стовпця. З різниць a_{ij} складають матрицю залишків $||a_{ij}||$. До неї дописують стовпець найбільш великих різниць $\mathrm{e_{ir}}$. Вибирають такі альтернативи $\mathrm{E_{i0}}$, в чиїх рядках знаходиться найбільш мале для такого стовпця значення [5].

Далі коротко розглянемо найпопулярніші комбіновані і розширені критерії прийняття рішень, котрі використовують для розв’язання погано структурованих проблем за умов невизначеності та ризику. Головна їх перевага полягає в тому, що під час розв’язання комбінуються одразу кілька підходів.

Отже, коротко оглянемо розширені критерії:

- критерій Баєса-Лапласа (Вальда) – його ідея полягає в тому, щоб обрати таку альтернативу, котра матиме найбільш великий середній виграш. Цей критерій максимізує ефективність, яка очікується;
- критерій Баєса-Лапласа (Севіджа) – головна ідея такого критерію полягає в оцінці ймовірних втрат для кожного варіанту і обранні такого варіанту, котрий матиме наймешну імовірну втрату;
- критерій добутків – орієнтований на величини виграшів. У результаті розрахунків обирають той варіант, який містить найбільший добуток.

Отже, у цьому підрозділі було розглянуто різні критерії теорії прийняття рішень, які використовують при розв’язанні погано структурованих проблем. Розглянуто такі критерії: Гермейера, Ходжа-Лемана, Вальда, Байєса-Лапласа, Гурвиця, Севіджа. А також розширені критерії: критерій Баєса-Лапласа (Вальда), критерій Баєса-Лапласа (Севіджа), критерій добутків. Окрім цього було з’ясовано поняття критерію. Використовуючи критерії прийняття рішень можливо формалізувати задачі, котрим притаманні низька визначеність, часткова кількість вхідної інформації, неоднозначність тощо.

Матеріал, який було розміщено у поточному розділі є важливим для розуміння наявних інструментів теорії прийняття рішень для розв’язання погано структурованих проблем за обставин ризику та невизначеності, а також для програмних реалізацій у четвертому розділі.

3 ОГЛЯД ЗАДАЧ ДЛЯ РЕАЛІЗАЦІЇ У ПРОГРАМНОМУ СЕРЕДОВИЩІ

Провідна ідея даного розділу полягає в огляді трьох погано структурованих проблем, котрі будуть розв’язані за допомогою програмних реалізацій у наступному розділі, тобто у четвертому.

Будуть описані такі задачі:

- задача про вибір найбільш доцільної стратегії для підприємства у період кризових економічних умов;
- задача про вибір оптимального місця для будівництва укриття або пункту незламності;
- задача про планування маршрутів для безпечного проїзду вантажівок.

Сформульовані задачі обиралися, виходячи з їх актуальності у сучасному сьогоденні. Розлянемо їх більш докладно.

3.1 Задача про вибір найбільш доцільної стратегії для підприємства у період кризових економічних умов

Ідеєю даного підрозділу є опис погано структурованої проблеми, Також наведення відповідної мети, імовірних стратегій розв’язку проблеми. Окрім цього наводяться пояснення, стосовно того, з яких причин задача є погано структурованою.

Розглянемо умову задачі.

Деяке підприємство, котре локаційно розташоване в одному з міст на території України працює під час повномаштабної війни. У такий складний час завод зіштовхується з певними ризиками. Наприклад, проблеми з логістикою, економічна нестабільність, зниження популярності продукції серед клієнтів, можливі удари ворога, ймовірні пожежі тощо. Необхідно

обрати найбільш доцільну стратегію, серед інших. Наприклад, перенесення виробництва у іншу зону, зменшення кількості витрат та робітників, зупинення підприємства та інші.

З'ясуємо, у зв'язку з якими причинами дана задача є погано структурованою:

- як відомо, ситуація у країні постійно змінюється, тому майже неможливо передбачити ймовірні втрати через удари ворога та пожежі після них;
- багато суперечливих факторів чинять певний вплив на підприємство, котрі потрібно враховувати одночасно;
- постійні ризики із зовнішнього середовища.

Розкриємо мету задачі.

Мета полягає у виборі найбільш оптимального варіанту, за оцінкою критерія Гурвиця, шляхом розробки програмного продукту.

3.2 Задача про вибір оптимального місця розташування для будівництва укриття або пункту незламності

Мета цього підрозділу полягає в описі погано структурованої проблеми, тобто задачі. Окрім цього, описується мета, імовірні варіанти рішення проблеми. Разом з цим наводяться пояснення, стосовно того, чому задача є погано структурованою.

Оглянемо умову задачі.

Місцева влада в Україні планує створеня укриття чи пункту незламності.

Припустимо, імовірні локації в межах міста наступні:

- локація, близька до житла;
- локація, близька до інфраструктури;
- локація, котра економічно вигідна.

Ця задача є погано структурованою через наявність постійних ризиків із зовнішнього середовища та суперечливих цілей.

Мета задачі полягає у виборі найбільш вдалої локації, за оцінкою критерія Вальда, шляхом розробки програми.

3.3 Задача про планування маршрутів для безпечного проїзду вантажівок

У цьому підрозділі наводиться опис погано структурованої проблеми, точніше задачі. Також розкриваються мета, ймовірні варіанти рішення проблеми. Разом з цим надаються пояснення, стосовно того, через що задача є погано структурованою.

Умова задачі.

Під час війни необхідно здійснити доставку певних життєво необхідних вантажів у зони, котрі локаційно розташовані в Україні. Такі зони можуть бути наближеними до лінії бойових зіткнень чи знаходитись на нещодавно деокупованих територіях.

З’ясуємо, з яких причин дана задача є погано структурованою:

- відсутність повної інформації, наприклад, про актуальність маршутів;
- мінлива ситуація в країні, через яку важко прогнозувати наслідки.

Мета задачі.

Зважаючи на все вище сказане, потрібно спланувати маршрут до зон таким чином, щоб він найбезпечнішим та найбільш швидким. Спланований маршрут обиратиметься за оцінкою Байєса-Лапласа.

Таким чином, у цьому розділі було розглянуто задачі, які актуальні сьогодні. Для цих задач було оглянуто їх умови, мету, а також надано короткі пояснення, з яких причин задачі вважаються погано структурованими. Для них наводитиметься програмна реалізація вже у наступному розділі.

4 АВТОМАТИЗАЦІЯ ДОСЛІДЖЕННЯ ТА РОЗВ'ЯЗАННЯ ПРАКТИЧНИХ ПОГАНО СТРУКТУРОВАНИХ ПРОБЛЕМ

У даному розділі демонструються три програмні реалізації до вище наведених погано структурованих задач. За кожною з програм проводиться тестування за трьома прикладами вхідних даних. Також у даному розділі уточнюються інструменти, за використанням яких були розроблені програмні продукти.

4.1 Засоби реалізації розроблюваних програмних продуктів

У наведеному підрозділі обґрунтовується вибір інструментів, за допомогою яких здійснювалися програмні реалізації задач з попереднього розділу.

Перш за все, визначимось з мовою програмування. Мовою, котрою буде зручно писати програми обрано мову програмування Python. Вона проста, має досить зрозумілий синтаксис та підходить для розв'язання завдань такого типу.

Визначимось із середовищем розробки. Середовищем розробки програмних продуктів до задач, наведених у попередньому розділі було обрано Visual Studio Code, адже це зручне, безкоштовне середовище.

З метою демонстрації вірної роботи програм та обчислень, у підрозділах 4.2, 4.3, 4.4 викладено результати тустування програм за трьома різними випадками вхідних даних.

4.2 Програмна реалізація задачі про вибір найбільш доцільної стратегії для підприємства у період кризових економічних умов

Мета даного підрозділу полягає в огляді створеного програмного коду до задачі 3.1 попереднього розділу з використанням критерію Гурвиця. Також тут наводяться результати компіляції програми для різних вхідних даних, покроковий алгоритм роботи коду та надається інструкція для користувача, яка буде корисною при використанні створеного програмного продукту.

Нижче наведемо лістинг програми до задачі про обрання найзбалансованішої стратегії для підприємства під час кризових економічних умов.

```
def input_matrix():
    while True:
        try:
            m = int(input("\nВведіть кількість стратегій: "))
            n = int(input("Введіть кількість сценаріїв: "))
            if m <= 0 or n <= 0:
                print("Розмірність має бути більша за нульову")
                continue
            break
        except ValueError:
            print("Необхідно вести ціле число")
    print("\nВведіть значення матриці виграшів (оцінок):")
    matrix = []
    for i in range(m):
        row = []
        print(f"Стратегія {i + 1}:")
        for j in range(n):
            while True:
```

```
            try:
                value = float(input(f"  Сценарій  {j + 1}: "))
                row.append(value)
                break
            except ValueError:
                print("Введіть число")
        matrix.append(row)
    return matrix

def main():
    print("Задача про вибір найбільш доцільної стратегії для підприємства у період кризових економічних умов")
    matrix = input_matrix()

    while True:
        try:
            alpha = float(input("\nВведіть коефіцієнт оптимізму α (від 0 до 1): "))
            if 0 <= alpha <= 1:
                break
            else:
                print("Значення має бути в межах [0, 1]")
        except ValueError:
            print("Введіть десяткове число")
    print("\nМатриця виграшів:")
    for row in matrix:
        print(" | ".join([f"{val:.2f}" for val in row]))

    results = []
    for i, row in enumerate(matrix):
        pessimistic = min(row)
```

```
        optimistic = max(row)
        gurvitz = alpha * optimistic + (1 – alpha) * pessimistic
        results.append(gurvitz)
        print(f"\nСтратегія {i + 1}: мінімум = {pessimistic}, максимум = {optimistic}, критерій Гурвиця = {round(gurvitz, 2)}")
    best_index = results.index(max(results))
    print(f"\nНайкраща стратегія: Стратегія {best_index + 1} (критерій = {round(results[best_index], 2)})")

if __name__ == "__main__":
    main()
```

Проведемо тестування програмного коду. При цьому, вхідні дані візьмемо для трьох різних варіантів матриць рішень:

$$H_1 = \begin{pmatrix} 0 & 10 & -5 \\ 2 & 3 & 8 \end{pmatrix}; \tag{4.1}$$

$$H_2 = \begin{pmatrix} 10 & 4 & 3 & -8 \\ -15 & 17 & 2 & 0 \\ 9 & 1 & -6 & 0 \\ 7 & 8 & 3 & -4 \end{pmatrix}; \tag{4.2}$$

$$H_3 = \begin{pmatrix} 5 & 8 & 9 & -15 \\ -2 & 24 & 6 & 14 \end{pmatrix}. \tag{4.3}$$

Нижче подано рисунки, що демонструють успішну роботу програми для різних матриць рішень (4.1) – (4.3), що наведені раніше.

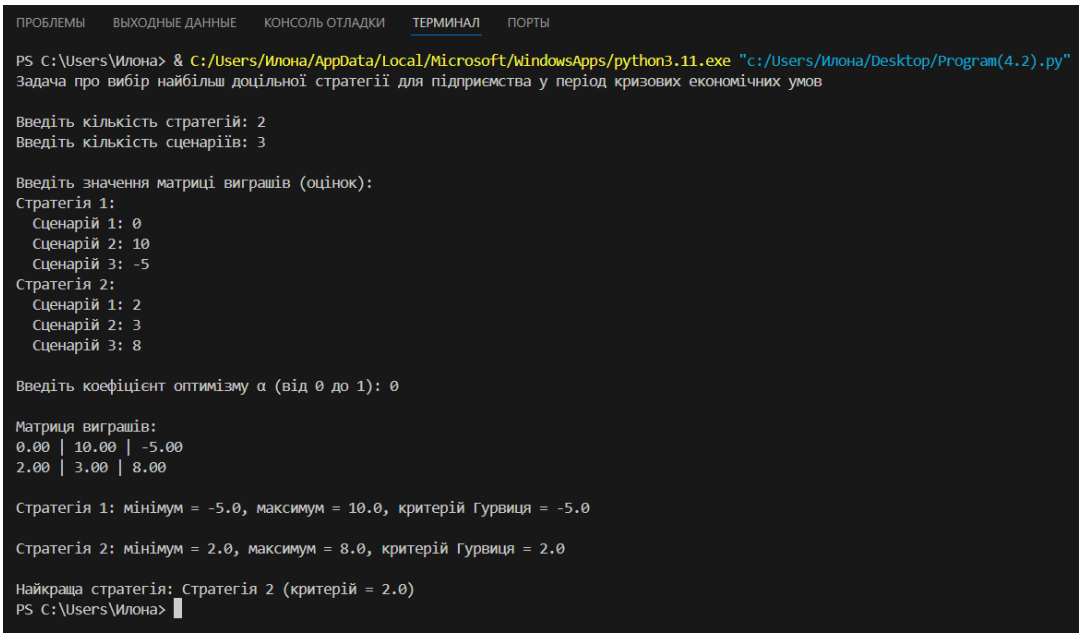


Рисунок 4.1 – Результат роботи програми для матриці рішень (4.1)

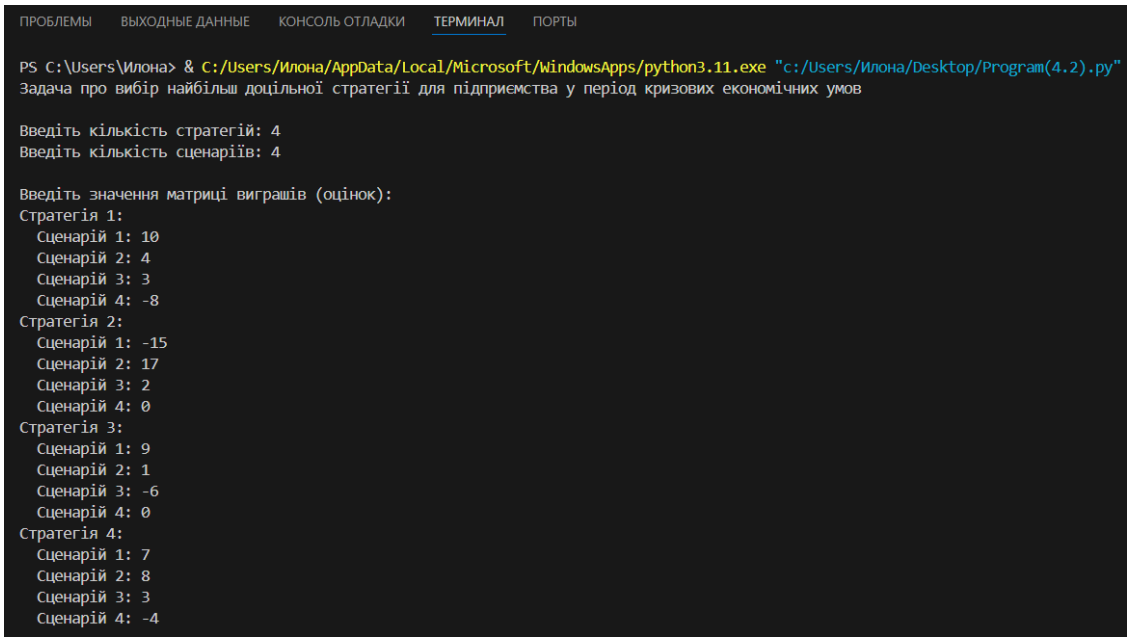


Рисунок 4.2 – Результат роботи програми для матриці рішень (4.2), аркуш 1

```
ПРОБЛЕМЫ    ВЫХОДНЫЕ ДАННЫЕ    КОНСОЛЬ ОТЛАДКИ    ТЕРМИНАЛ    ПОРТЫ

Введіть коефіцієнт оптимізму α (від 0 до 1): 1

Матриця виграшів:
  Сценарій 3: 2
  Сценарій 4: 0
Стратегія 3:
  Сценарій 1: 9
  Сценарій 2: 1
  Сценарій 3: -6
  Сценарій 4: 0
Стратегія 4:
  Сценарій 1: 7
  Сценарій 2: 8
  Сценарій 3: 3
  Сценарій 4: -4

Введіть коефіцієнт оптимізму α (від 0 до 1): 1

Матриця виграшів:
10.00 | 4.00 | 3.00 | -8.00
-15.00 | 17.00 | 2.00 | 0.00
9.00 | 1.00 | -6.00 | 0.00
7.00 | 8.00 | 3.00 | -4.00

Стратегія 1: мінімум = -8.0, максимум = 10.0, критерій Гурвиця = 10.0

Стратегія 2: мінімум = -15.0, максимум = 17.0, критерій Гурвиця = 17.0

Стратегія 3: мінімум = -6.0, максимум = 9.0, критерій Гурвиця = 9.0

Стратегія 4: мінімум = -4.0, максимум = 8.0, критерій Гурвиця = 8.0

Найкраща стратегія: Стратегія 2 (критерій = 17.0)
PS C:\Users\Илона>
```

Рисунок 4.2, аркуш 2

```
ПРОБЛЕМЫ    ВЫХОДНЫЕ ДАННЫЕ    КОНСОЛЬ ОТЛАДКИ    ТЕРМИНАЛ    ПОРТЫ

PS C:\Users\Илона> & C:/Users/Илона/AppData/Local/Microsoft/WindowsApps/python3.11.exe "c:/Users/Илона/Desktop/Program(4.2).py"
Задача про вибір найбільш доцільної стратегії для підприємства у період кризових економічних умов

Введіть кількість стратегій: 2
Введіть кількість сценаріїв: 4

Введіть значення матриці виграшів (оцінок):
Стратегія 1:
  Сценарій 1: 5
  Сценарій 2: 8
  Сценарій 3: 9
  Сценарій 4: -15
Стратегія 2:
  Сценарій 1: -2
  Сценарій 2: 24
  Сценарій 3: 6
  Сценарій 4: 14

Введіть коефіцієнт оптимізму α (від 0 до 1): 5
Значення має бути в межах [0, 1]

Введіть коефіцієнт оптимізму α (від 0 до 1): 0.4

Матриця виграшів:
5.00 | 8.00 | 9.00 | -15.00
-2.00 | 24.00 | 6.00 | 14.00

Стратегія 1: мінімум = -15.0, максимум = 9.0, критерій Гурвиця = -5.4

Стратегія 2: мінімум = -2.0, максимум = 24.0, критерій Гурвиця = 8.4

Найкраща стратегія: Стратегія 2 (критерій = 8.4)
PS C:\Users\Илона>
```

Рисунок 4.3 – Результат роботи програми для матриці рішень (4.3)

У результаті тестування розробленої програми (рис. 4.1–4.3), було з’ясовано, що код компілюється вірно та без помилок. Результатом роботи програми є можливість введення користувачем різних даних та вибір програмою оптимальної стратегії за критерієм Гурвиця.

Задля кращого розуміння архітектури наведеного програмного продукту, розглянемо його покроковий алгоритм роботи, який є інструкцією для кінцевого користувача:

- програма починається із запрошення введення розмірності майбутньої матриці. Необхідно ввести кількість стратегій, тобто кількість рядків матриці та ввести кількість сценаріїв, тобто кількість стовпців;
- далі програма пропонує ввести значення матриці виграшів, тобто оцінити наскільки гарний виходить результат виконання певної стратегії за певним сценарієм;
- потім програма пропонує ввести коефіцієнт оптимізму (від 0 до 1), де 0 – це песимістична позиція, а 1 – це оптимістична позиція;
- програма обчислює критерій Гурвиця за формулами (2.13) та (2.14);
- після цього програма виявляє таку стратегію, котра має найбільше значення критерію Гурвиця;
- останній етап полягає у виведенні на екран остаточно сформованої матриці та рекомендованої стратегії.

4.3 Програмна реалізація задачі про вибір оптимального місця для будівництва укриття або пункту незламності

Головною ідеєю цього підрозділу є огляд розробленого програмного коду до задачі 3.2 з третього розділу із застосуванням критерію Вальда. Ще тут наводяться результати компіляції програми для різних вхідних даних,

покроковий алгоритм роботи програми, а також надається інструкція користувача, яка буде корисною при використанні написаної програми.

Нижче наведемо лістинг програми до задачі про обрання найзбалансованішої стратегії для підприємства під час кризових економічних умов.

```
def input_matrix():
    while True:
        try:
            m = int(input("\nВведіть кількість локацій: "))
            n = int(input("Введіть кількість критеріїв оцінювання: "))
            if m <= 0 or n <= 0:
                print("Розмірність має бути більшою за нуль")
                continue
            break
        except ValueError:
            print("Введіть ціле число")

    print("\nВведіть оцінки для кожної локації:")
    matrix = []
    for i in range(m):
        row = []
        print(f"Локація {i + 1}:")
        for j in range(n):
            while True:
                try:
                    value = float(input(f"  Критерій {j + 1}: "))
                    row.append(value)
                    break
                except ValueError:
```

```
                print("Введіть число")
        matrix.append(row)
    return matrix

def main():
    print("Задача про вибір оптимального місця для будівництва укриття або пункту незламності")
    matrix = input_matrix()

    print("\nМатриця оцінок:")
    for row in matrix:
        print(" | ".join([f"{val:.2f}" for val in row]))

    min_values = [min(row) for row in matrix]
    for i, min_val in enumerate(min_values):
        print(f"Локація {i + 1}: мінімальна оцінка = {min_val}")

    best_index = min_values.index(max(min_values))
    print(f"\nНайкраща локація (за критерієм Вальда): Локація {best_index + 1} (оцінка = {min_values[best_index]})")

if __name__ == "__main__":
    main()
```

Проведемо тестування програмного коду. При цьому, вхідні дані візьмемо для трьох різних варіантів матриць рішень:

$$W_1 = \begin{pmatrix} 10 & 0 & -5 & 4 \\ 8 & 25 & 14 & -10 \end{pmatrix}; \quad (4.4)$$

$$W_2 = \begin{pmatrix} 15 & 4 \\ -10 & 2 \\ 4 & 0 \end{pmatrix}; \quad (4.5)$$

$$W_3 = \begin{pmatrix} 4 & 0 & 9 \\ 15 & -8 & -7 \\ 0 & 10 & 25 \\ 4 & 3 & 7 \\ -9 & 2 & 0 \end{pmatrix}. \quad (4.6)$$

Нижче подано рисунки, що демонструють успішну роботу програми для різних матриць рішень (4.4) – (4.6), що наведені раніше.

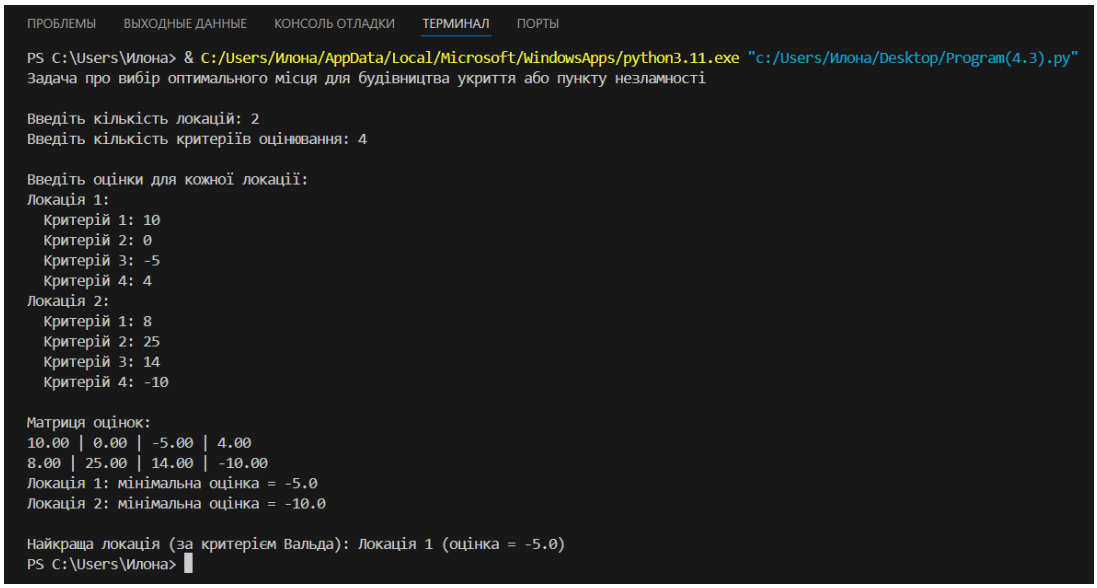


Рисунок 4.4 – Результат роботи програми для матриці рішень (4.4)

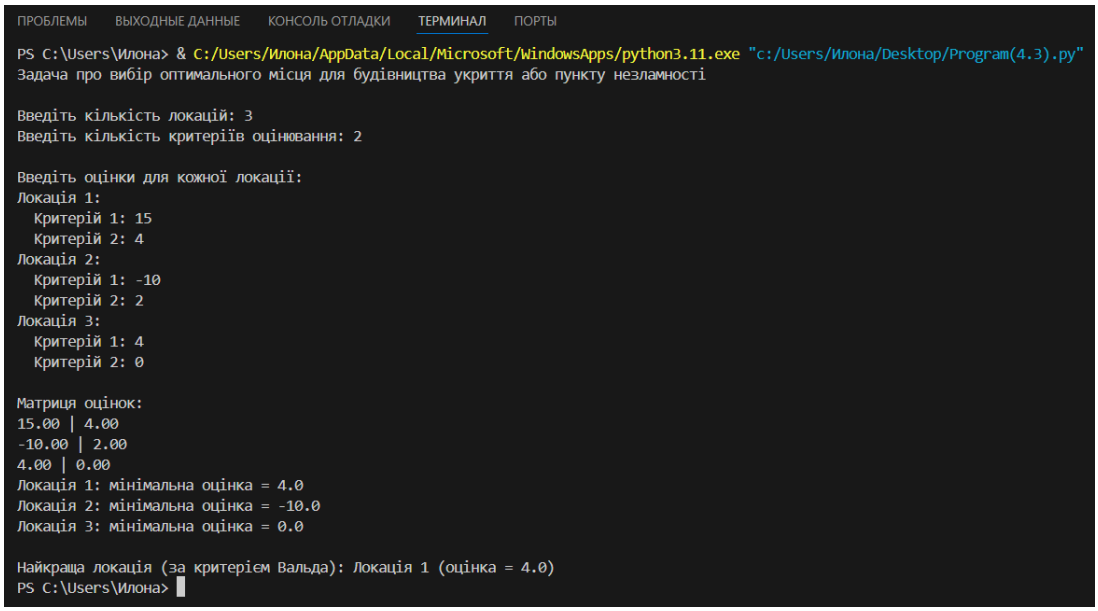


Рисунок 4.5 – Результат роботи програми для матриці рішень (4.5)

```
PS C:\Users\Илона> & C:/Users/Илона/AppData/Local/Microsoft/WindowsApps/python3.11.exe "c:/Users/Илона/Desktop/Program(4.3).py"
Задача про вибір оптимального місця для будівництва укриття або пункту незламності

Введіть кількість локацій: 5
Введіть кількість критеріїв оцінювання: 3

Введіть оцінки для кожної локації:
Локація 1:
  Критерій 1: 4
  Критерій 2: 0
  Критерій 3: 9
Локація 2:
  Критерій 1: 15
  Критерій 2: -8
  Критерій 3: -7
Локація 3:
  Критерій 1: 0
  Критерій 2: 10
  Критерій 3: 25
Локація 4:
  Критерій 1: 4
  Критерій 2: 3
  Критерій 3: 7
Локація 5:
  Критерій 1: -9
  Критерій 2: 2
  Критерій 3: 0

Матриця оцінок:
4.00 | 0.00 | 9.00
15.00 | -8.00 | -7.00
0.00 | 10.00 | 25.00
4.00 | 3.00 | 7.00
-9.00 | 2.00 | 0.00
Локація 1: мінімальна оцінка = 0.0
Локація 2: мінімальна оцінка = -8.0
Локація 3: мінімальна оцінка = 0.0
Локація 4: мінімальна оцінка = 3.0
Локація 5: мінімальна оцінка = -9.0

Найкраща локація (за критерієм Вальда): Локація 4 (оцінка = 3.0)
PS C:\Users\Илона>
```

Рисунок 4.6 – Результат роботи програми для матриці рішень (4.6)

У результаті тестування розробленої програми (рис. 4.5-4.7), було з’ясовано, що код компілюється вірно та без помилок. Результатом роботи програми є можливість введення користувачем різних даних та вибору оптимальної стратегії за критерієм Вальда.

Задля кращого розуміння архітектури наведеного програмного продукту, розглянемо його покроковий алгоритм роботи, який є інструкцією кінцевого користувача:

- програма починається з запрошення введення розмірності майбутньої матриці. Необхідно ввести кількість стратегій, тобто кількість рядків матриці та ввести кількість сценаріїв, тобто кількість стовпців;
- програма обчислює критерій Вальда за формулами (2.7) та (2.8).

- далі програма виявляє таку стратегію, котра має найбільше мінімальне значення критерію Вальда;
- останній етап полягає у виведенні на екран остаточно сформованої матриці та рекомендованої стратегії.

4.4 Програмна реалізація задачі про планування маршрутів для безпечного проїзду вантажівок

У цьому підрозділі наводиться створений програмний код до задачі 3.3 минулого розділу з використанням критерію Байєса-Лапласа. Окрім цього, тут надаються результати компіляції програми для різних вхідних даних, алгоритм роботи коду та подається інструкція для користувача, яка буде корисною при використанні розробленої програми.

Нижче наведемо лістинг програми до задачі про обрання найзбалансованішої стратегії для підприємства під час кризових економічних умов.

```
def input_matrix():
    while True:
        try:
            m = int(input("\nВведіть кількість маршрутів: "))
            n = int(input("Введіть кількість сценаріїв: "))
            if m <= 0 or n <= 0:
                print("Числа повинні бути додатними.")
                continue
            break
        except ValueError:
            print("Введіть цілі числа.")
```

```
    print("\nВведіть значення матриці:")
    matrix = []
    for i in range(m):
        row = []
        print(f"Маршрут {i + 1}:")
        for j in range(n):
            while True:
                try:
                    val = float(input(f"  Сценарій {j + 1}: "))
                    row.append(val)
                    break
                except ValueError:
                    print("Введіть число")
        matrix.append(row)
    return matrix

def laplace_criterion(matrix):
    scores = []
    for i, row in enumerate(matrix):
        avg = sum(row) / len(row)
        scores.append(avg)
        print(f"Маршрут {i+1}: середнє значення = {round(avg, 2)}")
    return scores

def main():
    print("Задача про планування маршрутів для безпечного проїзду
вантажівок")
    matrix = input_matrix()

    print("\nМатриця введена користувачем:")
```

```
    for row in matrix:
        print(" | ".join(f"{val:.2f}" for val in row))
    scores = laplace_criterion(matrix)
    best_index = scores.index(max(scores))
    print(f"\nНайбільш вдалий маршрут — Маршрут {best_index + 1} (оцінка = {round(scores[best_index], 2)})")

if __name__ == "__main__":
    main()
```

Проведемо тестування програмного коду. При цьому, вхідні дані візьмемо для трьох різних варіантів матриць рішень:

$$C_{BL1} = \begin{pmatrix} 5 & 8 \\ -12 & 21 \\ 19 & -5 \\ 0 & 1 \\ 4 & 2 \end{pmatrix}; \quad (4.7)$$

$$C_{BL2} = \begin{pmatrix} 8 & 9 & -2 & 3 & 4 \\ 5 & 7 & 12 & 28 & 9 \end{pmatrix}; \quad (4.8)$$

$$C_{BL3} = \begin{pmatrix} 7 & 9 & -10 & 12 \\ 2 & 4 & 0 & 6 \\ 4 & 2 & -8 & -9 \\ 15 & -5 & 2 & 0 \end{pmatrix}. \quad (4.9)$$

Нижче подано рисунки, що демонструють успішну роботу програми для різних матриць рішень (4.7) – (4.9), що наведені раніше.

```
ПРОБЛЕМЫ   ВЫХОДНЫЕ ДАННЫЕ   КОНСОЛЬ ОТЛАДКИ   ТЕРМИНАЛ   ПОРТЫ

PS C:\Users\Илона> & C:/Users/Илона/AppData/Local/Microsoft/WindowsApps/python3.11.exe "c:/Users/Илона/Desktop/Program(4.4).py"
Задача про планування маршрутів для безпечного проїзду вантажівок

Введіть кількість маршрутів: 5
Введіть кількість сценаріїв: 2

Введіть значення матриці:
Маршрут 1:
  Сценарій 1: 5
  Сценарій 2: 8
Маршрут 2:
  Сценарій 1: -12
  Сценарій 2: 21
Маршрут 3:
  Сценарій 1: 19
  Сценарій 2: -5
Маршрут 4:
  Сценарій 1: 0
  Сценарій 2: 1
Маршрут 5:
  Сценарій 1: 4
  Сценарій 2: 2

Матриця введена користувачем:
5.00 | 8.00
-12.00 | 21.00
19.00 | -5.00
0.00 | 1.00
4.00 | 2.00
Маршрут 1: середнє значення = 6.5
Маршрут 2: середнє значення = 4.5
Маршрут 3: середнє значення = 7.0
Маршрут 4: середнє значення = 0.5
Маршрут 5: середнє значення = 3.0

Найбільш вдалий маршрут — Маршрут 3 (оцінка = 7.0)
PS C:\Users\Илона>
```

Рисунок 4.7 – Результат роботи програми для матриці рішень (4.7)

```
PS C:\Users\Илона> & C:/Users/Илона/AppData/Local/Microsoft/WindowsApps/python3.11.exe "c:/Users/Илона/Desktop/Program(4.4).py"
Задача про планування маршрутів для безпечного проїзду вантажівок

Введіть кількість маршрутів: 2
Введіть кількість сценаріїв: 5

Введіть значення матриці:
Маршрут 1:
  Сценарій 1: 8
  Сценарій 2: 9
  Сценарій 3: -2
  Сценарій 4: 3
  Сценарій 5: 4
Маршрут 2:
  Сценарій 1: 5
  Сценарій 2: 7
  Сценарій 3: 12
  Сценарій 4: 28
  Сценарій 5: 9

Матриця введена користувачем:
8.00 | 9.00 | -2.00 | 3.00 | 4.00
5.00 | 7.00 | 12.00 | 28.00 | 9.00
Маршрут 1: середнє значення = 4.4
Маршрут 2: середнє значення = 12.2

Найбільш вдалий маршрут — Маршрут 2 (оцінка = 12.2)
```

Рисунок 4.8 – Результат роботи програми для матриці рішень (4.8)

У результаті тестування розробленої програми (рис. 4.7–4.9) було з’ясовано, що код компілюється вірно та без помилок. Результатом роботи програми є можливість введення користувачем різних даних та вибору оптимальної стратегії за критерієм Баєса-Лапласа.

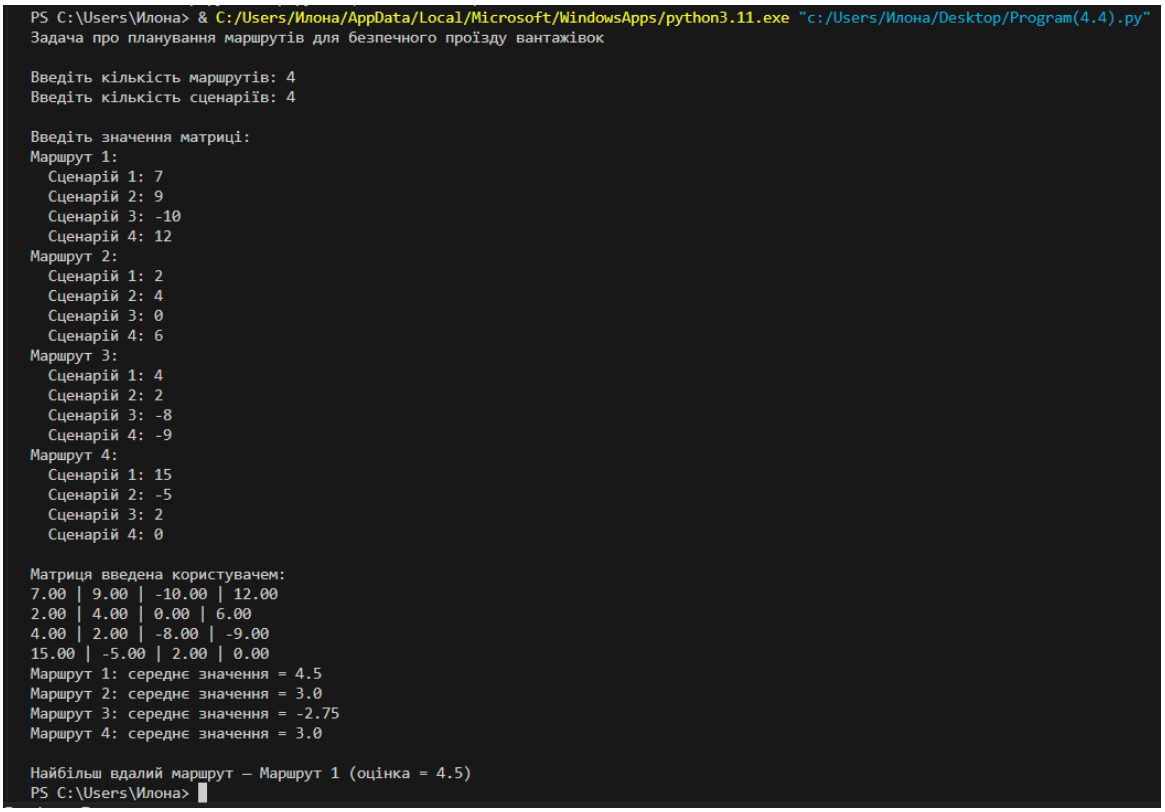


Рисунок 4.9 – Результат роботи програми для матриці рішень (4.9)

Задля кращого розуміння архітектури наведеного програмного продукту, розглянемо його покроковий алгоритм роботи, котрий є інструкцією кінцевого користувача:

– програма починається з запрошення введення розмірності майбутньої матриці. Необхідно ввести кількість стратегій, тобто кількість рядків матриці та ввести кількість сценаріїв, тобто кількість стовпців;
– програма обчислює критерій Байєса-Лапласа за формулами (2.10) та (2.11);
– далі програма виявляє таку стратегію, котра має найбільше мінімальне значення критерію Байєса-Лапласа;
– останній етап полягає у виведенні на екран остаточно сформованої матриці та рекомендованої стратегії.

Отже, у цьому розділі було наведено програмні реалізації задач, які були описані у третьому розділі. Продемонстровано по три різні приклади до кожної із задач. Описано інструменти, за допомгою яких були створені

наведені програми. Наведено алгоритми роботи кодів, які фактично є інструкціями для користувача.

Можна зробити висновок, що у результаті практичного дослідження, яке виконувалось шляхом написання програмних продуктів було успішно розв’язано конкретні практичні погано структуровані проблеми. Для розв’язання проблем такого типу використовувалися інструменти теорії прийняття рішень, тобто критерії Гурвиця, Вальда та Байєса-Лапласа.

ВИСНОВКИ

У кваліфікаційній роботі бакалавра за допомогою використаних джерел проаналізовано погано структуровані проблеми за використанням методів теорії прийняття рішень, розроблено програмні продукти для автоматизації процесу розв'язання погано структурованих проблем.

Структурно робота поділяється на чотири розділи.

У першому розділі наведено різноманітні теоретичні відомості з теорії прийняття рішень, представлено класифікацію проблем, а особлива увага приділено погано структурованим проблемам, надано постановку задачі прийняття рішень в умовах невизначеності та ризику, розгянуто розподіли задач прийняття рішень за певними ознаками.

У другому розділі проаналізовано особливості методів теорії прийняття рішень та їх використання при розв'язанні погано структурованих проблем, описані критерії теорії прийняття рішень.

Третій розділ присвячений опису задач, розв'язок котрих буде реалізований у наступному розділі шляхом написання відповідних програм.

У четвертому розділі наведено програмні реалізації практичних погано структурованих проблем, що були сформульовані та описані у попередньому розділі, за допомогою критеріїв Гурвиця, Вальда та Байєса-Лапласа. За кожною з розроблених програм проведено тестування за трьома різними вхідними даними для кожної задачі. Разом із цим висвітлено покрокові алгоритми роботи програм, які фактично є інструкціями для кінцевого користувача.

Проаналізувавши результати дослідження, можна зробити наступні висновки:

– у результаті написання кваліфікаційної роботи бакалавра було досягнуто встановленої мети, тобто проаналізовано погано структуровані проблеми інструментами теорії прийняття рішень,

розроблено програмне забезпечення для автоматизації процесу розв’язання проблем поганої структурованості;

- обґрунтовано особливості використання критеріїв теорії статистичних рішень як ефективних методів для розв’язання погано структурованих проблем;
- погано структуровані проблеми, що були розв’язані в ході дослідження, демонструють успішне застосування методів теорії статистичних рішень – за допомогою класичних критеріїв прийняття рішень;
- здійснено програмні реалізації практичних задач із з використанням критерів Гурвиця, Вальда та Байєса-Лапласа;
- тестування розроблених програм показало високу надійність та точність обчислень. Усі результати тестових прикладів відповідали очікуваним значенням, що свідчить про правильність реалізованих алгоритмів.

ПЕРЕЛІК ПОСИЛАНЬ

1. Кушлик-Дивульська О. І., Кушлик Б. Р. Основи теорії прийняття рішень. Київ, 2014. 94 с.

2. Горда О. В. Теорія прийняття рішень : конспект лекцій. Київ : КНУБА, 2023. 144 с.

3. Бутко М. П., Мащенко В. П. Теорія прийняття рішень. Київ : Центр учбової літератури, 2018. 360 с.

4. Пономарьов В. І., Лимаренко О. І. Теорія прийняття рішень : навч. посіб. Київ : НАУ, 2011. 264 с.

5. Застосування методології теорії прийняття рішень до слабо структурованих проблем URL : https://drive.google.com/file/d/1akzNHeoRUydSVdXwkPFjPXPFYe5WB4mj/view?usp=sharing (дата звернення : 03.04.2025).

6. Основні поняття та визначення теорії прийняття рішень (ТПР) : лекція. URL : https://learn.ztu.edu.ua/pluginfile.php/371456/mod_resource/content/1/1-2.%D0%A2%D0%9F%D0%A0_%D0%9B%D0%B5%D0%BA%D1%86%D1%96%D1%8F%201-2%20(1).pdf (дата звернення : 05.04.2025).

7. Системний аналіз і прийняття рішень : презентація. URL : https://en.ppt-online.org/492976 (дата звернення : 10.04.2025).

8. Анікін В. К., Крилов Є. В., Пасько В. П. Теорія прийняття рішень : конспект лекцій. Київ : КПІ ім. Ігоря Сікорського, 2023. 80 с. URL : https://files.znu.edu.ua/files/Bibliobooks/Inshi78/0058523.pdf (дата звернення : 09.04.2025).

9. Жураковська О. С. Теорія прийняття рішень : навч. посіб. для студ. спец. 126 «Інформаційні системи та технології» та спец. 121 «Інженерія програмного забезпечення». Київ : КПІ ім. Ігоря Сікорського, 2020. 99 с. URL : https://ela.kpi.ua/server/api/core/bitstreams/deaf214b-73d1-447f-8e2c-7ce1417e6d83/content (дата звернення : 25.04.2025).

10. Мельничук О. С. Словник іншомовних слів. 1-е вид. Київ : Українська радянська енциклопедія, 1974. 865 с.

11. Simon H. A. The Structure of Ill Structured Problem // Artificial Intelligence. 1973. № 4. P. 181–201. URL: https://doi.org/10.1016/0004-3702(73)90011-8 (дата звернення: 10.04.2025).

12. Мащенко С. О. Конспект лекцій з теорії прийняття рішень [Електронний ресурс]. Київ : КНУ ім. Тараса Шевченка. URL: https://csc.knu.ua/media/study/asp/decision_theory_mashchenko/lecture/lec1.pdf (дата звернення: 16.06.2025).

Декларація
академічної доброчесності
здобувача ступеня вищої освіти ЗНУ

Я, Лапенко Ілона Олексіївна
студентка 4 курсу, денної форми навчання, математичного факультету, спеціальності 113 прикладної математики, адреса електронної пошти ilona.lapenko2013yandex.ru@gmail.com, – підтверджую, що написана мною кваліфікаційна робота бакалавра на тему «Погано структуровані проблеми та їх дослідження засобами теорії прийняття рішень» відповідає вимогам академічної доброчесності та не містить порушень, що визначені у ст. 42 Закону України «Про освіту», зі змістом яких ознайомлений/ознайомлена;

– заявляю, що надана мною для перевірки електронна версія роботи є ідентичною її друкованій версії;

– згоден/згодна на перевірку моєї роботи на відповідність критеріям академічної доброчесності у будь-який спосіб, у тому числі за допомогою інтернет-системи, а також на архівування моєї роботи в базі даних цієї системи.

Студент

		І. О. Лапенко
(дата)	(підпис)	(ініціали, прізвище)

Науковий керівник

		В. В. Леонтьєва
(дата)	(підпис)	(ініціали, прізвище)