

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ПРИКЛАДНОГО
СИСТЕМНОГО АНАЛІЗУ
Кафедра математичних методів системного аналізу

До захисту допущено:
Завідувач кафедри
Оксана ТИМОЩУК
«__» _____ 2024 р.

ДИПЛОМНА РОБОТА
на здобуття ступеня бакалавра
за освітньо-професійною програмою «Системний аналіз і управління»
спеціальності 124 «Системний аналіз»
на тему: «Теоретико-ігровий підхід в задачі вибору локації»

Виконав:
студент IV курсу, групи КА-03
Ліфтеров Валентин Олександрович _____

Керівник:
доцент, к.ф.-м.н. Барановська Леся Валеріївна _____

Консультант з нормоконтролю:
к.ф.-м.н. Статкевич Віталій Михайлович _____

Консультант з економічного розділу:
доцент, к.е.н., Рощина Надія Василівна _____

Рецензент:
доцент, к.т.н., Харченко Костянтин Васильович _____

Засвідчую, що у цій дипломній роботі немає
запозичень з праць інших авторів без
відповідних посилань.
Студент _____

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Навчально-науковий інститут прикладного системного аналізу
Кафедра математичних методів системного аналізу

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 124 «Системний аналіз»

Освітня програма «Системний аналіз і управління»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Оксана ТИМОЩУК

«__» _____ 2024 р.

ЗАВДАННЯ

на дипломну роботу студенту

Ліфтерову Валентину Олександровичу

1. Тема роботи «Теоретико-ігровий підхід в задачі вибору локації», керівник роботи Барановська Леся Валеріївна доцент, к.ф.-м.н., затверджені наказом по університету від «__» _____ 2024 р. № _____.
2. Термін подання студентом роботи 11.06.2024 р.
3. Вихідні дані до роботи: матриця яка містить дані для побудови графів для вирішення задачі з вибору локації
4. Зміст роботи: Дослідження історії виникнення задач з вибору локації та їх сучасні приклади.

5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо):
презентація.
6. Консультанти розділів роботи.

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний	доцент, к.е.н., Рощина Н.В.		

7. Дата видачі завдання _____

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1.	Вивчення літератури за темою роботи.	18.04.2024	Виконано
2.	Підготовка першого розділу.	25.04.2024	Виконано
3.	Підготовка другого розділу.	01.05.2024	Виконано
4.	Початок розробки програмного продукту.	05.05.2024	Виконано
5.	Підготовка третього розділу.	29.05.2024	Виконано
6.	Підготовка економічної частини.	31.05.2024	Виконано
7.	Оформлення розділів відповідно до нормконтролю.	01.06.2024	Виконано
8.	Підготовка презентації доповіді.	05.06.2024	Виконано
9.	Оформлення дипломної роботи.	07.06.2024	Виконано

Студент

Валентин ЛІФТЕРОВ

Керівник

Леся БАРАНОВСЬКА

РЕФЕРАТ

Дипломна робота: 83 с., 17 рис., 6 табл., 2 додатки, 13 джерел.
ТЕОРЕТИКО-ІГРОВИЙ ПІДХІД В ЗАДАЧІ ВИБОРУ ЛОКАЦІЇ.

Об'єкт дослідження – теоретико-ігровий підхід в задачах вибору локації. Формалізації таких задач.

Предмет дослідження – практичне використання теоретико-ігрового підходу у вирішенні щоденних викликів різноманітних галузей діяльності людини.

Мета роботи – освоїти теоретико-ігровий підхід у задачах з вибору локації та розробити програмний продукт який вирішує задачу з вибору локації базового рівня.

Методи дослідження – аналіз відкритих наукових джерел, дослідження вирішення сучасних проблем вибору локації.

Актуальність – вирішення проблем вибору локації у стані високої урбанізованості сучасних міст.

Результати роботи – проведено дослідження того як виник теоретико-ігровий підхід в задачі вибору локації та створення програмного продукту який вирішує базову задачу відповідного типу.

Шляхи подальшого розвитку предмету дослідження – використання відкритих геоданих для перенесення задачі з уявної площини графів до реальної карти міста чи будь-якої іншої місцевості.

ABSTRACT

Diploma thesis: 83 p., 17 fig., 6 tables, 2 appendices, 13 sources.

GAME-THEORETIC APPROACH TO THE LOCATION SELECTION PROBLEM

Object of research – game-theoretic approach to location selection problems. Formalization of such tasks.

Subject of study – practical use of the theoretical game approach in solving daily challenges in various fields of human activity.

Purpose of study – master the game-theoretic approach to location selection tasks and develop a software product that solves the problem of basic level location selection.

Research methods – analysis of open scientific sources, research on solving modern problems of location selection.

Relevance – solving the problems of location selection in a state of high urbanization of modern cities.

Results – a study of how the theoretical and game approach to the problem of location selection and the creation of a software product that solves the basic problem of the corresponding type.

Ways to further develop the subject of research – the use of open geodata to transfer the problem from the imaginary plane of graphs to a real map of a city or any other area.

ЗМІСТ

ВСТУП.....	5
РОЗДІЛ 1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ.....	6
1.1 Історія виникнення задачі. Опис задачі.	6
1.2 Аналіз розташування в роботі екстрених служб. Вирішення задач TPLP та MTPLP в контексті роботи екстрених служб.	13
1.3 Локаційний аналіз в енергетичних системах. Енергетична доктрина при вирішенні задач TPLP, MTPLP	20
1.4 Висновки до розділу 1.....	23
РОЗДІЛ 2 ТЕОРЕТИКО-ІГРОВІ ІНСТРУМЕНТИ ДЛЯ ВИРІШЕННЯ ЗАДАЧ ЛОКАЦІЙНОГО АНАЛІЗУ. ПРИКЛАДИ ЗАСТОСУВАННЯ ПРИНЦИПІВ ТЕОРІЇ ІГОР ДЛЯ ВИРІШЕННЯ ЗАДАЧ ЛОКАЦІЙНОГО АНАЛІЗУ	24
2.1 Вступ.....	24
2.2. Теорія кооперативних ігор	25
2.2.1 Розподіл вигащів	27
2.2.2 Домінування розподілів вигащів	28
2.3 Теорія некооперативних ігор	29
2.4 Висновки до розділу 2.....	30
РОЗДІЛ 3 РОЗРОБКА КОМП'ЮТЕРНОЇ РЕАЛІЗАЦІЇ ДЛЯ РІШЕННЯ ЗАДАЧІ ПРО ДВА ПОШТОВИХ ВІДДІЛЕННЯ.....	31
3.1.Постановка задачі.....	31
3.2 Опис використаних бібліотек та методів	32
3.2.1 Опис основних змінних	33
3.3. Опис методу	34
3.4.Приклади роботи програмного продукту	35
3.5.Висновки до розділу 3.....	45
РОЗДІЛ 4 ФУНКЦІОНАЛЬНО-ВАРТІСНИЙ АНАЛІЗ ПРОГРАМНОГО ПРОДУКТУ.....	46
4.1. Формування завдання проєктування	46

	4
4.2. Обґрунтування функцій програмного продукту	47
4.3. Обґрунтування системи параметрів програмного продукту.....	52
4.4. Аналіз експертного оцінювання параметрів.....	54
4.5. Аналіз рівня якості варіантів реалізації функцій	58
4.6. Економічний аналіз варіантів розробки ПП	59
4.7. Вибір кращого варіанта техніко-економічного рівня	63
4.8. Висновки до 4 розділу.....	64
ВИСНОВКИ.....	65
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	66
ДОДАТОК А. ТЕКСТ ПРОГРАМИ.....	68
ДОДАТОК Б. ІЛЮСТРАТИВНИЙ МАТЕРІАЛ	77

ВСТУП

З давніх давен задача вибору локації стала однією з ключових у житті більшості людей на землі. З появою потреби переміщувати велику кількість людей, продовольства, товарів на відчутні відстані, контролювати колонізовані території. Пошук найбільш ефективних методів задовольнити свої логістичні потреби став буденністю. Першим з'явився стратегічний підхід до прокладання доріг, після чого морська торгівля та водні маршрути стали одним з ключових етапів розвитку світової торгівлі. Звичайно технологічний прогрес не стояв на місці і з появою залізних доріг, винайденню літаків і гелікоптерів світ логістики змінився на довгі роки вперед.

Це все безперечно вплинуло на якість та швидкість задоволення логістичних потреб людства. Але не все залежить від технічних засобів подолання логістичних маршрутів, сама їх структура, почат, розміщення точок зміни типу маршруту, розміщення логістичних хабів, - все це грає на менш ключову роль у процесі вдоволення логістичних потреб.

Логістичні задачі такого типу вирішуються у більшості сучасних галузей, від базової роздрібною торгівлі до складних технологічних виробництв будь-якого типу, де розташування матеріалів та обладнання має прямий вплив на оптимізацію процесів. Найбільш очевидним і найбільш актуальним питанням наразі є задоволення логістичних потреба армії. Ефективність сил логістики має чи не ключовий вплив на сучасну війну на виживання і без перебільшення, їм доводиться виконувати надскладні логістичні задачі з динамічними умовами кожного дня.

В поточній роботі ми приділяємо особливу увагу задачам з вибору локації перевалочного пункту, а саме задачам MLTP (Multiple Location of Transfer Points), TPLP (Transfer Point Location Problem), FTPLP (Facility and Transfer Points Location Problem) і буде розглядати теоретико-ігрові інструменти які використовуються при їх вирішенні.

РОЗДІЛ 1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Історія виникнення задачі. Опис задачі.

В першу чергу визначимо події якого характеру призвели до виникнення задачі вибору локації у тому виді у якому ми їх знаємо. Проблеми розміщення об'єктів вивчають стратегії фізичного розташування різних об'єктів (або ресурсів), щоб мінімізувати витрати, пов'язані з виконанням певної кількості заявок на товари чи послуги, які представлені групою клієнтів. Центральне питання в проблемі вибору локації полягає в тому, як саме ми плануємо розташовувати ці об'єкти. Звідси випливають додаткові запитання: які локації доступні для розміщення? Як ми робимо вибір між цими доступними опціями[4]?

Відповівши на перше питання ми визначаємо місце розташування. Друге питання вимагає визначення об'єктивної функції і простору попиту, що стосується, наприклад, зведення до мінімум збитків, незручностей, витрат або максимізації прибутку і якості послуг, оскільки питання місця розташування можна назвати особливим класом проблем оптимізації. В складних ситуаціях потрібно одночасно тримати під контролем якомога більшу кількість критеріїв, в той час як у деяких контекстах мета може бути визначення лише за допомогою одного критерію. Але не треба забувати що можуть існувати ряд обмежень, як з точки зору попиту, так і пропозиції. В розподілі товарів і послуг та управлінні і плануванні виробництва, центральну роль грає рішення про розміщення. На рівні стратегічного планування, рішення такого типу можуть надавати довгостроковий вплив. Як в наукових працях, так і реальному житті існує велика кількість методів, моделей і систем підтримки

рішень які направлені на представлення та вирішення декількох варіантів і додатків завдань розміщення.[1]

До цього переліку можуть входити такі завдання[11]:

1. розміщення складських приміщень у межах виробничо-збутового ланцюга з метою мінімізації часу, необхідного для доставки товарів;
2. розташування об'єктів публічних послуг (наприклад, шкіл або лікарень) для забезпечення максимального охоплення населення;
3. локація потенційно шкідливих об'єктів для зберігання та виробництва небезпечних матеріалів (наприклад, звалища чи спалювальні заводи) з метою мінімізації їх впливу на населення;
4. встановлення транспортних засобів (станції, зупинки, парковки) для покращення загальної транспортної інфраструктури;
5. організація точок обслуговування клієнтів та автоматизованих систем для підвищення якості банківських та фінансових послуг.

Хоча описані вище завдання мають різноманітні цілі, вхідні дані та обмеження, всі вони представляють собою варіації проблеми вибору локації. Незважаючи на їхню практичну важливість і застосування у сучасному світі, можна простежити їхній історичний розвиток, починаючи з задач, які були сформульовані ще у 17-му столітті багатьма видатними математиками, одним з яких був П'єр Ферма[10].

Задачу пошуку найменшої суми евклідових відстаней, вважають одною з перших задач визначення локації. Її основну версію сформулював французький математик П'єр де Ферма. Вона мала такий вигляд: «Якщо задані три точки на площині, знайти четверту точку, таку, що сума відстаней до трьох заданих точок мінімальна»[10].

Математичний запис задачі Ферма.

Позначимо задані точки наступним $P_1 = (a_1, b_1)$, $P_2 = (a_2, b_2)$, $P_3 = (a_3, b_3)$. Шукану точку - $Q = (x, y)$, а суму відстаней до точки Q із заданих опишемо таким чином[10]:

$$f(Q) = d(Q, P_1) + d(Q, P_2) + d(Q, P_3),$$

де $d(Q, P_i)$ - евклідова відстань між невідомою точкою Q та однією із точок P . Таким чином, $f(Q)$ – шукана відстань, яка має бути оптимізованою (в даному випадку мінімізованою)[10].

Загадка походження першого розв'язку задачі Ферма про пошук точки з мінімальною сумою евклідових відстаней до трьох заданих точок досі залишається нерозгаданою серед істориків. Існує кілька теорій, одна з яких приписує перше рішення Еванджелісті Торрічеллі, тоді як інші джерела вважають, що першим був єзуїт Бонавентура Кавальєрі. Метод Торрічеллі полягав у наступному[10]:

1. Для заданого трикутника $P_1P_2P_3$ метод Торрічеллі передбачає конструювання рівностороннього трикутника на кожному з його ребер.
2. Далі, навколо кожного з цих новостворених рівносторонніх трикутників описується окружність.
3. Точка перетину цих трьох окружностей і буде шуканою точкою Q , яку також називають точкою Торрічеллі. Ця точка має унікальну властивість — вона є точкою, з якої сума відстаней до трьох вершин трикутника є мінімальною, і вона визначається за допомогою геометричних конструкцій, викладених в методі Торрічеллі.

Бонавентура Кавальєрі також зробив важливе спостереження в контексті задачі про мінімальну суму відстаней. Він помітив, що кути, утворені променями, що виходять із точки Q до вершин заданого трикутника $P_1P_2P_3$, є рівними між собою і кожен дорівнює 120° градусів.

Це означає, що точка Q , відома також як точка Торрічеллі або Ферма-Торрічеллі, знаходиться таким чином, що мінімізує сумарну відстань до трьох вершин трикутника, і при цьому утворює рівні кути 120° зі сторонами трикутника, на який вона "дивиться". Ці кути є ознакою оптимального розташування точки Q відносно даних трьох точок[11].

1. Конструкція рівносторонніх трикутників: На кожному ребрі заданого трикутника $P_1P_2P_3$ необхідно добудувати рівносторонній трикутник. Ці

- рівносторонні трикутники будуть мати сторони, які дорівнюють довжинам відповідних ребер $P_1P_2P_3$, та P_1P_2 оригінального трикутника.
2. Прокладення відрізків: Проведемо відрізки що з'єднують вершини рівносторонніх трикутників що тільки що були утворені та які не лежать на ребрах трикутника $P_1P_2P_3$, з протилежними вершинами оригінального трикутника $P_1P_2P_3$. Тобто, кожен такий відрізок з'єднує "віддалену" вершину рівностороннього трикутника, що базується на одному ребрі, з вершиною оригінального трикутника, яка не є частиною цього ребра.
 3. Знаходження точки Q : Точка перетину цих відрізків буде шуканою точкою Q . Ця точка, відома як точка Ферма-Торрічеллі, мінімізує суму відстаней до трьох вершин трикутника $P_1P_2P_3$.

Запропоновані Томасом Сімпсоном відрізки назвали на його честь - прямі Сімпсона. Але цим його вклад не обмежився, їм було запропоноване узагальнення для задачі Ферма яке в результаті було визнане дуже важливим. Він припустив що кожній вершині можна надати так названу "вагу", з чого випливає що задача з пошуку найкоротшої суми евклідових відстаней перетворюється у задачу з пошуку точки де буде найменша сума цих самих "вагів" відстаней для заданих вершин [1].

Так само, відомий математик Якоб Штейнер який прославився як засновник синтетичної геометрії кривих сформував трохи інше формулювання завдання розміщення: при заданих n точках, потрібно знайти мінімальне дерево яке може їх з'єднати. Пізніше це завдання буде розширене і стало відомим як завдання про мінімальний остовном дерева. Приклад з додатків: точки можна прийняти як представлення міста які потрібно з'єднати між собою залізничною або автодорожною мережею, або ж вузли в електричному ланцюзі [1].

Але на цьому розвиток задач з вибору локацій не зупинився, до прикладу американський економіст Гарольд Готелінг [8] ввів кардинально новий підтип задачі, суть якої була у розв'язку проблеми взаємного розміщення двох

торговців. Вони конкурували між собою надаваючи однакові послуги з ціллю отримати більшу частину попиту. У своїх роздумах Готелінг мав за належне те, що попит має одновимірний простір і по суті є просто прямою. Запропонована Готеллінгом модель виглядала так: зробимо припущення, що на пляжі розташовані два кіоски які продають морозиво. Кожен власник має право сам обрати місце розташування кіоску на пляжі, при цьому він не знав вибір конкурента. За умову бралось те, що клієнти розподілені по пляжу рівномірно і кожен клієнт буде обирати найближчий кіоск задля мінімізування відстані яку він має пройти пішки. Також, існувало обмеження, що довжина пляжу скінченна, щоб ні один з клієнтів не задумався про те, щоб уникнути купівлю морозива через те що найближчий кіоск задалеко. Спочатку це може здатися дивним. Очевидно що магазин отримає більше клієнтів, якщо він знаходиться далеко від свого конкурента? Відповідь на це питання здається тривіальною, але Готеллінг довів протилежне. Введемо Продавця А і Продавця Б. Обидва хочуть продати якомога більше морозива. Першого дня вони встановили свої кіоски на половині пляжу, продавець А - з одного боку, а продавець Б - з іншого. Таким чином, кожен з них отримує приблизно половину відвідувачів пляжу. Це справедливий розподіл. Однак продавець Б хоче більше. Тому на другий день він пересуває свою ятку на середину пляжу. Тепер він отримує більше клієнтів - близько 62,5%, залишаючи продавцю А лише 37,5%. Побачивши це, продавець А також переїжджає на середину пляжу на третій день. Тепер вони обидва повертаються до розподілу клієнтів 50/50. Жоден з них не може переїхати, не втративши клієнтів на користь іншого. З цього випливає що рівноважне рішення в умовах конкурентного ринку - це рішення, яке дозволяє розмістити обидва кіоску в центрі без урахування потреб клієнтів. Дана модель відома як "модель лінійного міста". Готеллінг буквально вивів формулу локальної рівноваги Неша яка є одним з ключових понять теорії ігор[10].

Але не дивлячись на, без перебільшень, видатну та довгу історію розвитку задач з вибору локації, їх розвиток не зупиняється до наших днів.

Одним з найбільш очевидних інструментів для вирішення завдань локалізації сьогодні можна назвати графи, але подібні завдання почали набувати популярність аж у 1960-х роках. У 1964 році, відомий американський математик Іранського походження - Луїс Хакімі змінив погляд людства на задачу Вебера[11]. Він перетворив її у задачу з графом і довів, що рішення для кожного завдання яке можна назвати оптимальним, забезпечується засобами які розташовані тільки в вузлах графа. Завдання p -медіани (назва задачі яка була запропонована Луїсом Хакімі) полягало у визначенні розташування p об'єктів і пов'язаних з ними клієнтів, з ціллю звести до мінімуму загальну ціну перевезення між клієнтами і об'єктами. [2]. Але Хакімі не зупинився на цьому і логічним продовженням його роботи було виведення так названої задачі p -центру. Її суть полягала у знаходженні на графіку кількості p -об'єктів, задля мінімізування відстані між довільним об'єктом та клієнтами, а саме від найбільш віддаленого клієнта до довільного об'єкта. Не дивлячись на те що задача була сформована самим Хакімі, загальний метод для її розв'язання був запропонований у 1978 році американським математиком Ізраїльського походження Зві Дрезнером та Джорджем Весоловським. Їх формулювання має такий вид[11]:

Розглянемо розробку нового механізму, призначеного для обслуговування n пунктів попиту. Можливе створення точок передачі, які об'єднують декілька пунктів попиту. Вартість пересування від такої точки передачі до кінцевого об'єкта розраховується з коефіцієнтом скорочення 1. Витрати на переміщення від пунктів попиту до точки передачі еквівалентні безпосереднім витратам на дорогу до об'єкта. В основі моделі лежить вибір не тільки кінцевого об'єкта, але й набору p перевалочних пунктів таким чином, щоб мінімізувати загальний час поїздок усіх клієнтів. В деяких заявках можуть бути вказані лікарні, які приймають пацієнтів через вертолітні майданчики. В першу чергу, пацієнтів доставляють на кареті швидкої до вертолітного майданчика, а звідти — швидше транспортуються до лікарні. Лікарня як кінцева точка вже визначена, але потребує встановлення місць для вертолітних

майданчиків. Загальна модель допомагає визначити оптимальне розташування як лікарні, так і вертолітних майданчиків[11].

У своєму дослідженні (яке було створено в співавторстві з американським математиком Ізраїльського походження Одедом Берманом) [3] вчені виділяють три підзадачі:

1. Проблема визначення точки передачі (англ. The Transfer Point Location Problem), TPLP

Для завдання, де визначено місцеположення кінцевого об'єкта і задано ряд точок, що мають бути пов'язані через одну точку передачі, наша мета полягає в тому, щоб визначити ідеальне місце для цієї точки передачі.

2. Множинне розташування пунктів передачі (англ. The Multiple Location of Transfer Points), MLTP

Модель MLTP (Multi-Level Transport Problem), яка вже розробляється для оптимізації транспортних потоків та розміщення перевалочних пунктів, також адаптована для визначення оптимального розташування кінцевого об'єкта. Цей підхід відомий як "загальна модель". Це дозволяє комплексно підійти до питання розміщення як перевалочних точок, так і кінцевих об'єктів, враховуючи множинність рівнів перевезень і відповідне їх координування для досягнення максимальної ефективності.

3. Проблема розміщення об'єктів і пунктів передачі (англ. The Facility and Transfer Points Location Problem), FTPLP

Фактично загальна модель яка дозволяє знайти оптимальне розташування об'єкта. Визначення точки передачі знаходить численні застосування в реальному світі. Теоретико-ігровий підхід, який використовує Берман, застосовується для оптимального розміщення медичних об'єктів, таких як лікарні з вертолітними майданчиками. Цей підхід включає в себе модель, де лікарня виконує роль центрального вузла для прийому пацієнтів, доставлених вертольотом, а потім перевезених каретами швидкої допомоги до пункту пересадки і далі до кінцевих медичних закладів.

Основні компоненти моделі:

1. Лікарня з вертолітним майданчиком: Лікарня обладнана для прийому пацієнтів, доставлених повітряним транспортом, де вони отримують первинну медичну допомогу.
2. Пункт пересадки: Це проміжна локація, де пацієнтів перевантажують з вертольота на карети швидкої допомоги. Цей пункт знаходиться поруч з лікарнею, забезпечуючи швидкий трансфер.
3. Карети швидкої допомоги: Використовуються для транспортування пацієнтів від пункту пересадки до спеціалізованих медичних закладів.

Принципи вибору локації:

1. Оптимізація відстаней: Мінімізація відстаней між пунктами транспортування для швидкого доставлення пацієнтів.
2. Врахування часу транспортування: Час є критичним фактором у медичних перевезеннях, тому розташування має забезпечувати найшвидший можливий шлях.
3. Ресурсоефективність: Оптимальне використання наявних ресурсів, таких як вертольоти та карети швидкої допомоги.

Цей підхід сприяє зниженню часу реакції та підвищенню шансів на успішне лікування, оскільки дозволяє швидко та ефективно транспортувати пацієнтів до потрібних медичних закладів. Загалом, цю модель можна застосувати скрізь, де відбувається збір або розповсюдження товарів.

1.2 Аналіз розташування в роботі екстрених служб. Вирішення задач TPLP та MTPLP в контексті роботи екстрених служб.

Аналіз місця розташування є критичним компонентом для ефективної роботи екстрених служб. Ці служби представляють собою складну систему, що включає організацію, транспортні та комунікаційні мережі, підготовлених фахівців і адміністраторів. Вони повинні працювати в координації для

забезпечення успішного реагування на надзвичайні ситуації. Планування таких систем потребує значних зусиль для забезпечення їх ефективності.

Однією з ключових задач є розподіл та розташування транспортних засобів, зокрема аварійних. Важливими критеріями при виборі місць для розміщення транспортних засобів є час реагування, доступність і покриття регіонів попиту. Регіони попиту розподіляються на основі ймовірності виникнення викликів. Попит у цих регіонах вважається випадковим і слідує часооднорідному пуассонівському процесу, що дозволяє прогнозувати середню кількість викликів у кожному регіоні за одиницю часу.

Кожен регіон має список транспортних засобів, відсортованих за близькістю. При надходженні виклику перевіряється наявність транспортних засобів, починаючи з найближчого. Якщо транспортний засіб доступний, він відправляється на виклик; якщо ні, виклик втрачається. Час обслуговування включає час на подорож до місця виклику, сам процес обслуговування і повернення до бази.

Для оптимізації розташування використовуються моделі покриття. Модель покриття множини визначає мінімальну кількість об'єктів для покриття всіх точок попиту, з метою зменшення витрат. Модель максимального покриття спрямована на максимізацію кількості обслуговуваних запитів при фіксованій кількості об'єктів. Основна різниця між цими моделями полягає у фокусі на витратах у першому випадку та на максимальному обслуговуванні у другому.

Такі моделі допомагають приймати стратегічні рішення щодо розміщення транспортних засобів, диспетчерської політики та оптимізації ресурсів, що забезпечує ефективну роботу екстрених служб і покращує якість обслуговування населення.

Ефективне вирішення задач TPLP та MTPLP вимагає використання сучасних математичних моделей і алгоритмів. Наприклад, моделі покриття (Coverage Models) дозволяють визначити мінімальну кількість точок розміщення, необхідних для забезпечення покриття всіх точок попиту. Моделі

максимального покриття (Maximum Coverage Models) спрямовані на максимізацію кількості обслуговуваних запитів при обмежених ресурсах. Використання цих моделей забезпечує баланс між витратами та ефективністю, дозволяючи екстреним службам надавати максимально можливий рівень обслуговування з мінімальними витратами.

Одним з прикладів дослідження щодо оптимізації розташування медичних вертолітних майданчиків під назвою "Modelling optimal location for pre-hospital helicopter emergency medical services"[4]. Це дослідження аналізує розташування медичних вертолітних майданчиків з урахуванням логістичних викликів і характеристик пацієнтів. Основною метою цього дослідження було визначити найкращі місця для розміщення медичних вертолітних майданчиків, щоб забезпечити максимальне покриття і мінімальний час реагування на виклики.

У дослідженні використовувалися дані про госпіталізацію внаслідок важких травм, зібрані з реєстру травм провінції Британська Колумбія. Ці дані включали інформацію про пацієнтів, які зазнали важких травм та потребували госпіталізації на три і більше днів. Географічна інформаційна система (ГІС) використовувалася для моделювання охоплення послугами медичних вертолітів, що дозволило точно визначити регіони з найбільшим попитом на ці послуги. Як приклад у дослідженні видно мапу де розгорнуто програму раннього реагування/автоматичного запуску служб повітряної швидкої допомоги для географічних районів, що охоплюють регіон Нижнього материка провінції та прилеглі до нього, а також окремі райони північної та центральної частини острова Ванкувер. (рис. 1.1.)

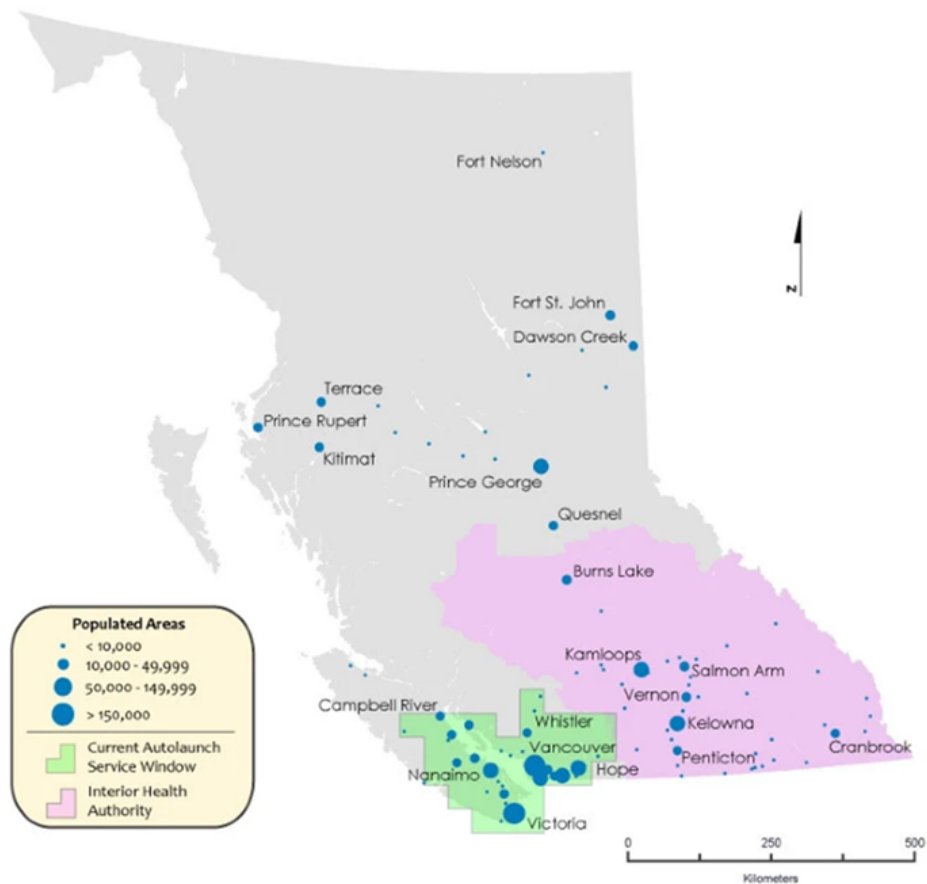


Рисунок 1.1 Карта зони реагування BCAS Autolaunch, яка зараз базується в Міжнародному аеропорту Ванкувера. [4]

На рисунку 1.2 показано карту МГВ та територію, на яку припадає одна година їзди дорогою, пов'язану з обома об'єктами, а також основні населені пункти по всій території. Зверніть увагу, що люди, які проживають за межами годинного часу в дорозі, є кандидатами на отримання ГЕМС.

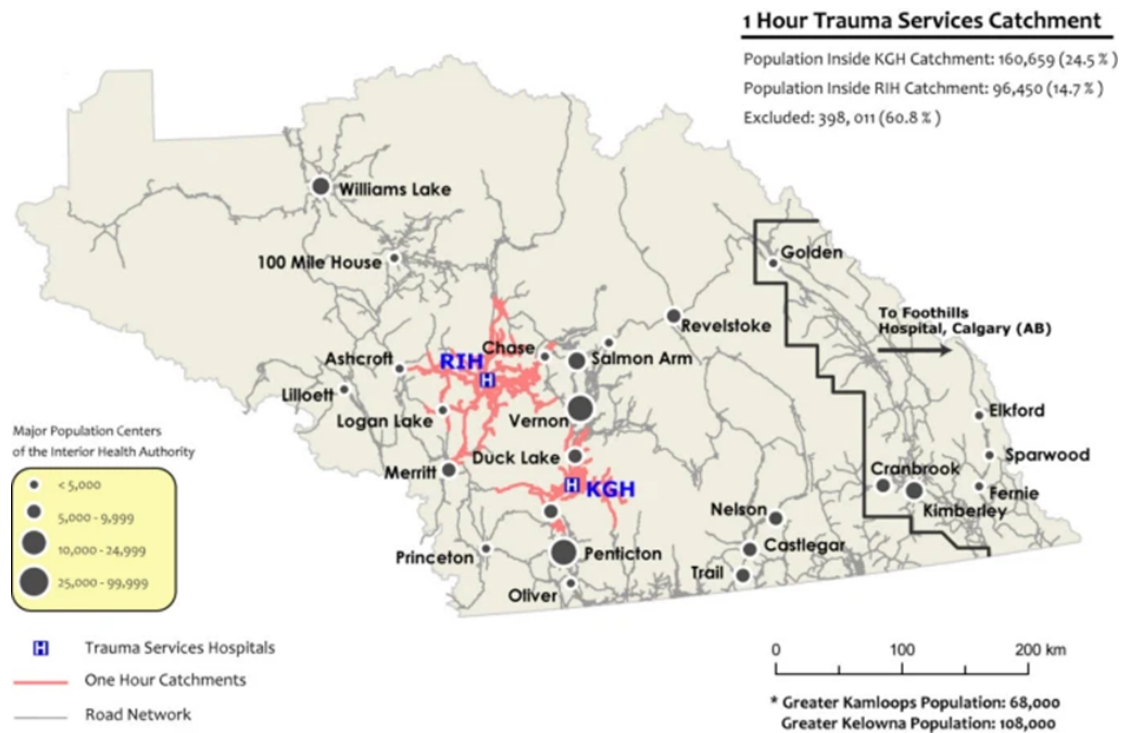


Рисунок 1.2 Існуючий час доїзду до двох центрів третинного рівня надання медичної допомоги в Управлінні охорони здоров'я внутрішніх справ. Все населення, яке проживає за межами однієї години їзди на автотранспорті, є кандидатами на отримання СМД. [4]

Дослідження показало, що оптимальне розташування вертолітних майданчиків може значно зменшити час реагування на виклики та покращити доступність медичних послуг. Наприклад, було встановлено, що розміщення майданчиків у стратегічних точках у центральному коридорі долини Оканаган дозволяє охопити більшість населення регіону в межах однієї години їзди. Це дозволяє забезпечити своєчасну медичну допомогу, що є критичним фактором для виживання пацієнтів з важкими травмами.

Для аналізу використовувалися методи просторової оптимізації, які враховували різні сценарії розвитку подій і дозволяли моделювати різні стратегії розміщення транспортних засобів. Результати дослідження підтвердили, що правильне планування розташування вертолітних майданчиків може суттєво покращити ефективність роботи екстрених

медичних служб і забезпечити більш ефективне реагування на надзвичайні ситуації.

Це дослідження ілюструє, як застосування математичних моделей та ГІС може допомогти у вирішенні задач TPLP та MTPRP, забезпечуючи більш ефективне розташування ресурсів екстрених медичних служб.

Ще одним прикладом є робота "A Multi-factor Spatial Optimization Approach for Emergency Medical Facilities in Beijing"[5]. Дане дослідження було проведено у Пекіні та було опубліковане у виданні International Journal of Geo-Information (MDPI), зосереджене на розробці мультифакторної просторової оптимізаційної моделі для покращення розташування екстрених медичних об'єктів у міському середовищі.

Основною метою цього дослідження було оптимізувати розташування екстрених медичних об'єктів у Пекіні з урахуванням таких факторів, як час, трафік, та демографічні характеристики населення. Враховуючи динамічні зміни у транспортній системі та різноманітні потреби населення, дослідники прагнули створити модель, яка б забезпечила швидкий та ефективний доступ до медичної допомоги в надзвичайних ситуаціях.

У дослідженні використовувалася геоінформаційна система (ГІС) та аналіз мережі для моделювання просторових характеристик екстрених медичних об'єктів. Було враховано тимчасові зміни у дорожній мережі, включаючи варіації швидкості руху в різні періоди доби та дні тижня. Аналіз також включав розподіл населення, рівень щільності населення, та середню кількість викликів на медичні послуги. Наприклад, було виділено розташування закладів медичної допомоги в Пекіні (рис. 1.3.)

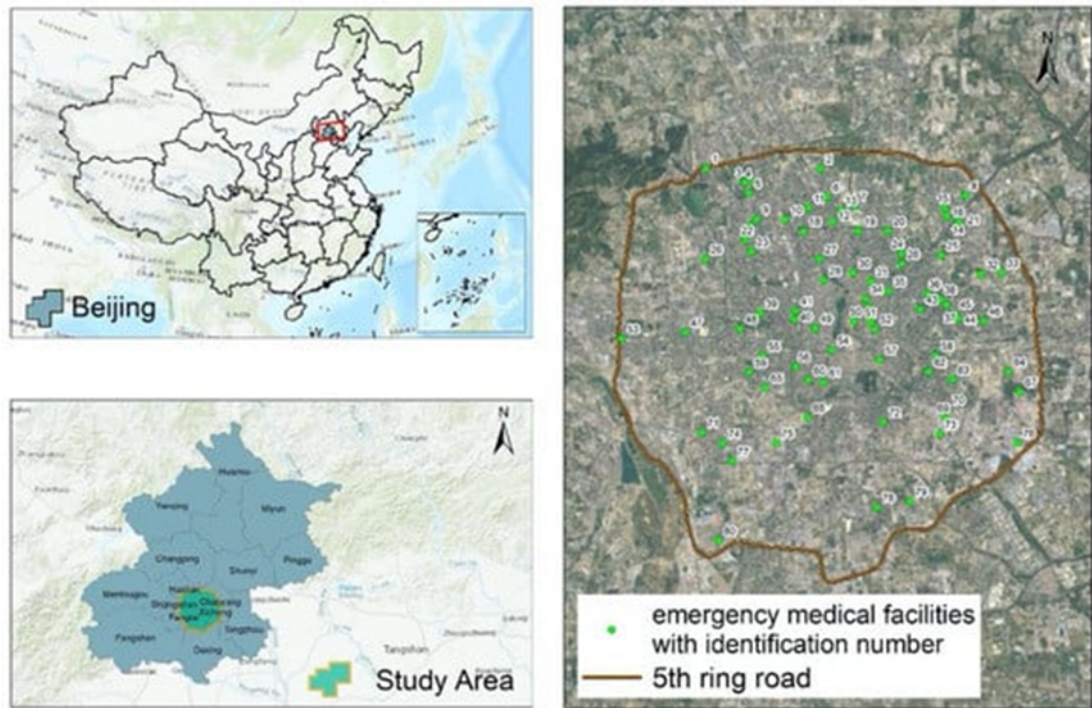


Рисунок 1.3 Територія дослідження та розташування закладів екстреної медичної допомоги в Пекіні. [5]

Дослідження виявило, що найвищий попит на екстрені медичні послуги спостерігається в центральних, північних та східних районах Пекіна протягом дня (з 8:00 до 20:00), та в центральних і північних районах уночі (з 20:00 до 8:00). (рис. 1.4., рис. 1.5.)

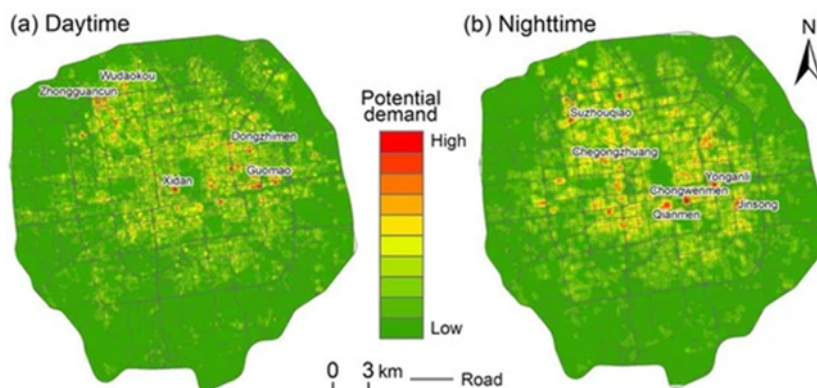


Рисунок 1.4. Просторовий розподіл потенційного попиту на екстрені послуги в різний час: (а) вдень і (б) вночі.[5]

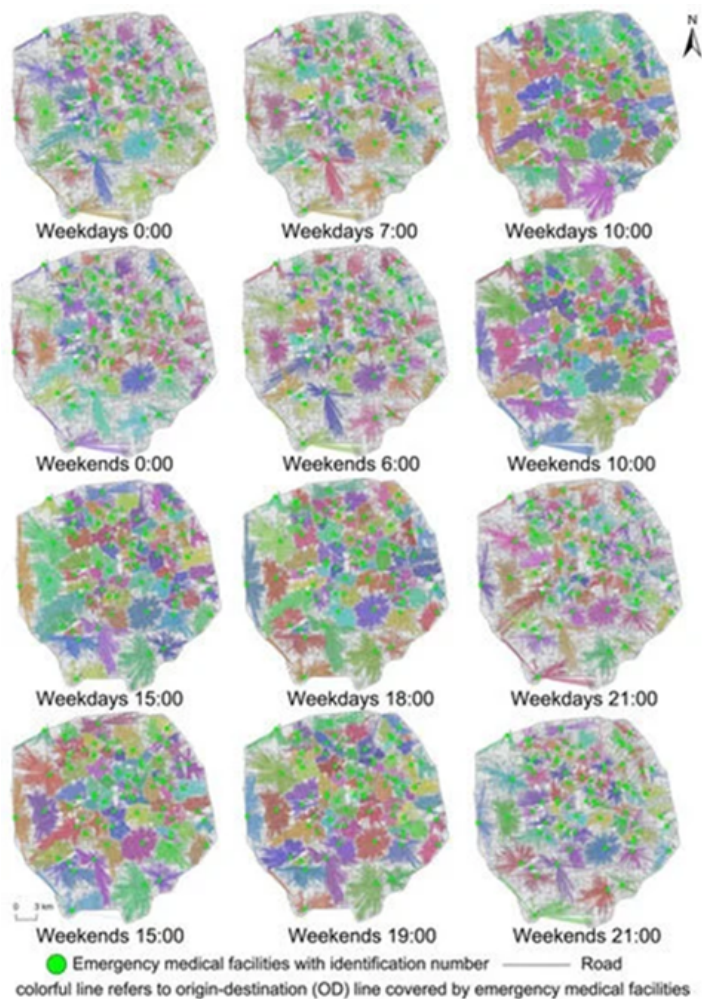


Рисунок 1.5. Розподіл покриття закладів екстреної медичної допомоги в різний час [5].

1.3 Локаційний аналіз в енергетичних системах. Енергетична доктрина при вирішенні задач TPLP, MTPLP

Локаційний аналіз відіграє ключову роль у плануванні та оптимізації розміщення об'єктів зеленої енергетики, таких як вітрові та сонячні електростанції. Задачі TPLP (Task Priority Location Problem) та MTPLP (Multi-Task Priority Location Problem) використовуються для визначення найбільш ефективних місць розташування цих об'єктів з урахуванням різних факторів, включаючи кліматичні умови, доступність інфраструктури та близькість до

споживачів енергії. Наприклад, розташування вітрових електростанцій вимагає детального аналізу вітрових ресурсів, топографії місцевості та віддаленості від мережевої інфраструктури для мінімізації витрат на транспортування енергії та максимізації її виробництва. Аналогічно, для сонячних електростанцій враховуються рівні сонячного випромінювання, наявність площ для встановлення панелей та їх орієнтація для забезпечення максимальної ефективності. Використання математичних моделей оптимізації дозволяє знайти компроміс між цими факторами та розробити стратегії, які забезпечать стабільне та ефективне енергопостачання з мінімальним впливом на навколишнє середовище. Це не лише сприяє досягненню цілей сталого розвитку, але й допомагає у зниженні вуглецевого сліду та підвищенні енергетичної незалежності регіонів.

Один з прикладів використання таких моделей є платформа REopt, розроблена Національною лабораторією відновлюваної енергії США (NREL)[6]. Ця платформа допомагає оптимізувати енергетичні системи для будівель, кампусів, громад та мікромереж, рекомендуючи оптимальне поєднання відновлюваної енергії, традиційного виробництва та технологій зберігання енергії для досягнення цілей економії витрат, стійкості, зниження викидів та покращення енергетичних характеристик.

Іншим дослідженням щодо оптимізації розміщення вітрових та сонячних електростанцій є робота, проведена в регіоні Шаньдун, Китай[7]. Дослідники використовували ГІС для інтеграції просторових даних про ресурси вітру та сонця, топографію місцевості, існуючу інфраструктуру та екологічні обмеження. Це дозволило створити детальну карту регіону з потенційними локаціями для розміщення електростанцій. Провінція Шаньдун була представлена 55 віртуальними майданчиками спостережень, розташованими в центрі сітки (рис. 1.6.). Для оцінки були використані 10-річні дані про швидкість вітру та сонячне випромінювання з 2006 по 2015 роки для цих віртуальних пунктів спостереження.

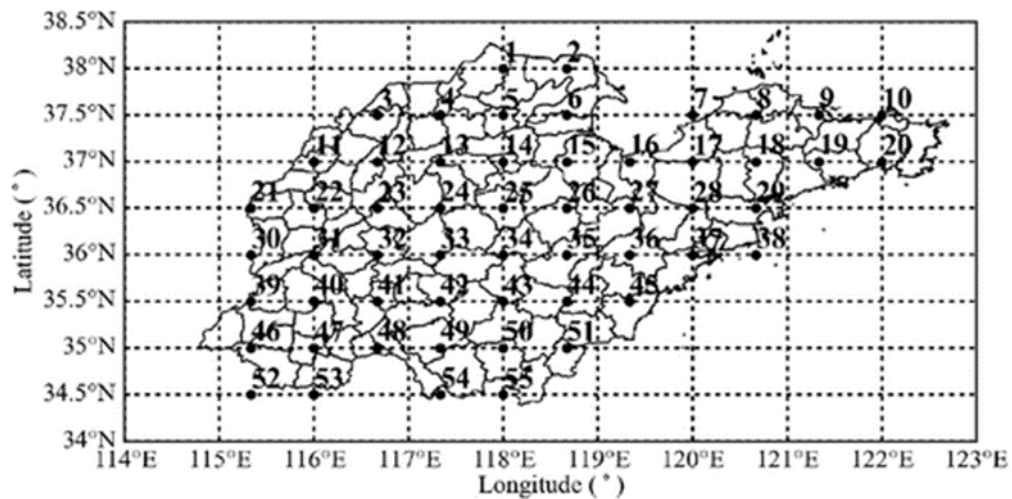


Рисунок 1.6. Віртуальні спостережні майданчики в провінції Шаньдун[7]

Математичні моделі оптимізації оцінювали різні сценарії розміщення, з урахуванням максимальної ефективності використання ресурсів та мінімізації витрат.

В результаті дослідження було визначено стратегічні точки для встановлення вітрових та сонячних електростанцій, що дозволило значно підвищити ефективність використання енергетичних ресурсів. Оптимізоване розміщення допомогло зменшити витрати на будівництво та експлуатацію електростанцій, а також забезпечило стабільне та ефективне енергопостачання регіону. Врахування місцевих екологічних обмежень також сприяло зниженню впливу на навколишнє середовище. Використання ГІС та оптимізаційних методів дозволило забезпечити максимальну ефективність та економічність проекту, сприяючи сталому розвитку та зменшенню вуглецевого сліду.

1.4 Висновки до розділу 1

В цьому розділі ми дослідили еволюцію задачі вибору локації та розглянули приклади її використання у контексті екстрених служб та енергетичних систем. Ми також обговорили роль локаційного аналізу у побудові критично важливої соціальної інфраструктури та сучасних енергетичних систем.

РОЗДІЛ 2 ТЕОРЕТИКО-ІГРОВІ ІНСТРУМЕНТИ ДЛЯ ВИРІШЕННЯ ЗАДАЧ ЛОКАЦІЙНОГО АНАЛІЗУ. ПРИКЛАДИ ЗАСТОСУВАННЯ ПРИНЦИПІВ ТЕОРІЇ ІГОР ДЛЯ ВИРІШЕННЯ ЗАДАЧ ЛОКАЦІЙНОГО АНАЛІЗУ

2.1 Вступ

Задачі вибору локацій виходять далеко за межі теорії ігор і їх вирішення доволі сильно інтегровано у повсякденність абсолютно різних прошарків населення, незалежно від роду їх діяльності. Як ми побачили на прикладах наведених у першому розділі, задачі вибору локації не обмежуються задачами в таких очевидних сферах як логістика чи торгівля.

Очевидним прикладом є задача оптимізації розміщення центрів розподілу товарів для великої логістичної компанії. Метою є мінімізація витрат на транспортування товарів від центрів розподілу до кінцевих споживачів. Важливо врахувати такі фактори, як відстань до споживачів, доступність транспортних шляхів та вартість оренди приміщень. Зазвичай центри розподілу розташовуються поблизу основних магістралей та транспортних вузлів, що дозволяє зменшити час доставки та знизити витрати.

Але крім вже наведених прикладів, зовсім нещодавно людство стояло перед екзистенційною проблемою у вигляді пандемії і задачу визначення найкращих місць для розташування пунктів вакцинації можна назвати однією з ключових у питанні вирішення даної кризи. Важливо врахувати густоту населення, доступність громадського транспорту, а також наявність достатньої кількості приміщень для організації пунктів вакцинації. Це дозволяє забезпечити максимальне охоплення населення вакцинацією та мінімізувати черги та час очікування.

У контексті розвитку зеленої енергетики актуальною є задача визначення місць для встановлення зарядних станцій для електромобілів. Враховуються такі фактори, як потоки транспорту, наявність паркувальних місць, близькість до житлових і комерційних зон, а також доступність електромережі. Правильне розташування зарядних станцій сприяє зручності користування електромобілями та підтримці сталого розвитку.

У цьому розділі ми розглянемо методи вирішення таких задач, опишемо математичні моделі та основні проблеми, з якими стикаються при роботі з цими методами. На конкретних прикладах продемонструємо, як ці методи працюють на практиці.

2.2. Теорія кооперативних ігор

Незважаючи на обширність теорії ігор як розділу математики, як і будь-яка наука вона має низку основних понять та об'єктів вивчення. Основним таким об'єктом є математичні моделі прийняття рішень в умовах конфлікту. Конфлікт у теорії ігор – це ситуація, в якій інтереси двох або більше учасників (гравців) протилежні, і кожен з них намагається максимізувати свої вигоди за рахунок інших. У такій ситуації рішення одного гравця впливає на результати інших гравців, що створює необхідність враховувати можливі стратегії опонентів при прийнятті власних рішень.

Кооперативні ігри в теорії ігор представляють собою сценарії, де гравці можуть укласти угоди та працювати разом для досягнення кращих результатів для всіх учасників. Класичною кооперативною грою можна назвати гру розподілу пирога, в якій декілька гравців розподіляють певний ресурс (наприклад, пиріг). Мета гри – знайти спосіб розподілу, який буде справедливим для всіх гравців. Це може включати домовленості про те, як поділити ресурс так, щоб кожен отримав задовільну частку.

Кооперативна гра в характеристичній формі визначається як пара (N, v) де:

1. $N = \{1, 2, \dots, n\}$ – скінченна множина гравців[12].
2. $v: 2^N \rightarrow \mathbf{R}$ – характеристична функція, що кожній коаліції $S \subseteq N$ ставить у відповідність число $v(S)$, яке відображає сумарний виграш, що може бути отриманий коаліцією S .

Функція v має задовольняти умову $v(\emptyset) = 0$.

Довільну підмножину $S \subset N$ будемо називати коаліцією в грі. Коаліцію $T = N \setminus S$, що є доповненням до множини всіх гравців, називатимемо додатковою коаліцією для коаліції $S \subset N$. Під стратегією x_S коаліції S розуміємо набір стратегій, обраних усіма гравцями, що входять до коаліції S . Речову функцію v , визначену на всіх коаліціях $S \subset N$ будемо називати характеристичною функцією гри з безліччю гравців N . При цьому для будь-яких коаліцій T, S , що не перетинаються ($T \subset N, S \subset N, S \cap T = \emptyset$) виконується нерівність[12]:

$$v(T) + v(S) \leq v(T \cup S),$$

факт що функція v має задовольняти умову $v(\emptyset) = 0$, залишається валідним.

Ця властивість означає, що можливості коаліції що об'єдналась завжди буде більші або рівні можливостям двох окремих коаліцій, які діють незалежно одна від одної. Отже, всі гравці мають стимул об'єднатися в максимальну коаліцію N . Дана функцію є суперадитивною.

Кооперативна гра в формі характеристичної функції має такий вигляд:

$$G(N, v), \text{ де } N \text{ – множина гравців, } v \text{ – характеристична функція[12]}$$

Розглянемо приклад кооперативної гри трьох компаній A, B і C , які планують спільний проект з будівництва інфраструктурного об'єкта, такого як міст. Вигоди від участі в проекті залежать від кількості і складу учасників коаліції. Нехай множина гравців $N = \{A, B, C\}$. Ця задача може бути описана грою трьох осіб у формі характеристичної функції.

Характеристична функція v визначає виграш для кожної коаліції:

1. $v(\emptyset) = 0$ – жодна компанія не отримує виграш, якщо не бере участі

2. $v(\{A\}) = 0, v(\{B\}) = 0, v(\{C\}) = 0$ – кожна окрема компанія не може самостійно реалізувати проект.
3. $v(\{A, B\}) = 10$ – компанії A і B можуть реалізувати частину проекту і отримати виграш 10.
4. $v(\{A, C\}) = 12$ – компанії A і C можуть реалізувати іншу частину проекту і отримати виграш 12.
5. $v(\{B, C\}) = 14$ – компанії B і C можуть реалізувати третю частину проекту і отримати виграш 14.
6. $v(\{A, B, C\}) = 25$ – всі три компанії об'єднуються і реалізують весь проект, отримуючи виграш 25.

Звідси ми бачимо, що можливості об'єднаної коаліції (всі гравці разом) завжди більші будь-якого поділу на окремі коаліції, що діють незалежно одна від одної. Це створює у всіх гравців стимул об'єднатися в максимальну коаліцію N .

2.2.1 Розподіл виграшів

Розподіл виграшів є ключовим питанням у теорії кооперативних ігор, оскільки він визначає, як сумарний виграш коаліції повинен бути поділений між її учасниками. Один з найважливіших підходів до розподілу виграшу є ядро гри. Ядро гри представляє множину розподілів, при яких жодна коаліція не може отримати більший виграш, ніж їй гарантує цей розподіл. Формально, розподіл виграшу $x = (x_1, x_2, \dots, x_n)$ належить ядру, якщо виконується рівність[10]:

$$\sum_{i \in N} x_i = v(N),$$

Це умова колективної раціональності. У випадку де вона не виконується, то або гравці ділять між собою виграш, який не можна реалізувати, або існує розподіл, при якому кожен гравець отримує більше, ніж його частка у цьому конкретному розподілі що робить сам поділ фізично неможливим. Таким чином, вектор $x = (x_1, x_2, \dots, x_n)$ лише за умови дотримання колективної раціональності є допустимим розподілом.

та нерівності:

$$x_i \geq v(\{i\}) \forall i \in N$$

Це умова індивідуальної раціональності. Вона означає, що кожен гравець гарантує собі отримання виграшу у разі участі у максимальній коаліції, при цьому цей виграш буде щонайменше дорівнює тому, що він міг би отримати, діючи самостійно, за умови відсутності підтримки інших гравців. де $v(\{i\})$ – виграш гравця i у випадку, якщо він діє самостійно.

Таким чином, $x = (x_1, x_2, \dots, x_n)$ є допустимим розподілом у кооперативній грі, якщо виконується умова колективної раціональності (сума виграшів дорівнює виграшу коаліції) та умова індивідуальної раціональності (кожен гравець отримує не менше, ніж міг би отримати самостійно).

2.2.2 Домінування розподілів виграшів

В теорії кооперативних ігор важливу роль відіграє поняття домінування розподілів виграшів. Воно дозволяє оцінити перевагу одного розподілу над іншим з точки зору задоволення коаліційних інтересів. Розуміння цього поняття є ключовим для аналізу стійкості та справедливості запропонованих розподілів виграшів.

Нехай $x = (x_1, x_2, \dots, x_n)$ та $y = (y_1, y_2, \dots, y_n)$ – два різних розподіли виграшів у кооперативній грі (N, v) . Говорять, що розподіл y у домінує розподіл x (або y є кращим за x), якщо для будь-якої коаліції $S \subseteq N$ виконується нерівність[10]:

$$\sum_{i \in S} y_i \geq v(S)$$

та існує хоча б одна коаліція $T \subseteq N$, для якої справджується строга нерівність:

$$\sum_{i \in T} y_i > v(T).$$

Домінування за Парето є окремим випадком домінування розподілів виграшів. Розподіл y називається Парето-домінуючим над розподілом x , якщо виконується умова:

$$\sum_{i \in N} y_i = \sum_{i \in N} x_i = v(N)$$

та для кожного гравця $i \in N$ справджується нерівність:

$$y_i \geq x_i$$

при цьому для хоча б одного гравця $j \in N$ виконується строга нерівність:

$$y_j > x_j$$

Парето-домінування означає, що при переході від розподілу x до y виграш жодного з гравців не зменшується, а для хоча б одного гравця збільшується.

2.3 Теорія некооперативних ігор

На відміну від теорії кооперативних ігор, теорія некооперативних ігор описує стратегічні взаємодії між раціональними гравцями, що приймають рішення самостійно, без формування коаліцій або укладання угод. Вона надає математичний апарат для аналізу конфліктних ситуацій, де кожен

гравець намагається максимізувати свій власний виграш, враховуючи можливі дії інших гравців.

Некооперативна гра в нормальній формі визначається як трійка (N, A, u) де [13]:

1. $N = \{1, 2, \dots, n\}$ – скінченна множина гравців.
2. $A = A_1 \times A_2 \times \dots \times A_n$ – множина стратегій, де A_i – множина стратегій гравця i .
3. $2u = (u_1, u_2, \dots, u_n)$ – вектор функцій виграшу, $u_i : A \rightarrow R$ i визначає виграш гравця i при наборі стратегій $a = (a_1, a_2, \dots, a_n)$.

Кожен гравець $i \in N$ вибирає стратегію $a_i \in A_i$ з метою максимізації свого виграшу u_i . Важливо враховувати, що виграш кожного гравця залежить не тільки від його власної стратегії, але і від стратегій інших гравців. Гра відбувається в такий спосіб. Теорія некооперативних ігор знаходить застосування в багатьох галузях, включаючи економіку, політичні науки, біологію та інформатику. Вона використовується для аналізу ринкової конкуренції, стратегічного планування, переговорів, військових конфліктів, а також у проектуванні алгоритмів і мереж.

2.4 Висновки до розділу 2

У цьому розділі ми розглянули основні постулати та математичний апарат теорії ігор, зокрема кооперативні та некооперативні ігри. Ми також обговорили методи вирішення задач на площині та в мережах, такі як задача мінімізації суми евклідових відстаней (MTPLP). До цього розділу були включені детальні пояснення та приклади для кращого розуміння застосування цих методів у практичних ситуаціях.

РОЗДІЛ 3 РОЗРОБКА КОМП'ЮТЕРНОЇ РЕАЛІЗАЦІЇ ДЛЯ РІШЕННЯ ЗАДАЧІ ПРО ДВА ПОШТОВИХ ВІДДІЛЕННЯ

3.1. Постановка задачі

В данному розділі ми розглянемо задачу, яку буде розв'язувати програмний продукт, а саме задача вибору локації двох поштових відділень. У теорії графів, галузі математики та інформатики, існує відома задача про маршрут Гуана, задача китайського листоноші, тур листоноші або задача перевірки маршруту полягає у знаходженні найкоротшого замкненого шляху або ланцюга, який відвідує кожне ребро (зв'язного) неорієнтовного графа принаймні один раз. Ми розглянемо програмну реалізацію відозміненої гри в основі якої лежить гра пол вибір локації двох ресторанів яка запропонована в 5 розділі книги «Теорія ігор в прикладах» (англ. «Game Theory Through Examples») [9]. де мета стратегічно визначити оптимальні місця для розташування поштових відділень у заданій мережі графа. Гравці обирають вузли, де вони хочуть розмістити поштові відділення, і їхні вибори впливають на витрати на маршрутизацію та загальні виграші на основі k-центрум задачі китайського листоноші.

Налаштування гри:

1. Мережевий граф: Граф, що складається з вузлів і ребер, представляє область, де можуть бути розташовані поштові відділення. Кожне ребро має пов'язану вагу, яка представляє вартість маршрутизації.
2. Гравці: У грі беруть участь два гравці, кожен з яких обирає вузол у графі як місце для свого поштового відділення.

3. Вибір: Гравці незалежно обирають вузли, де вони хочуть розмістити свої поштові відділення. Вибори можуть бути однаковими або різними для кожного гравця.

Потрібно розробити такий програмний продукт який досягає таких цілей:

1. Мінімізувати витрати на маршрутизацію: Гравці прагнуть обрати місця для поштових відділень, які мінімізують суму k найбільших витрат на маршрутизацію у мережі.
2. Оптимізувати виграші: Кожен гравець прагне максимізувати свій виграш, стратегічно обираючи місце для поштового відділення, або централізуючи, або диференціюючи свої вибори на основі структури мережі.

3.2 Опис використаних бібліотек та методів

Гра реалізована за допомогою Dash, веб-інструментарію для Python, який дозволяє створювати інтерактивні веб-додатки. Окрім Dash, у проекті використовуються декілька інших бібліотек для візуалізації мереж, обробки даних та роботи з графами.

1. Dash є основною бібліотекою, що використовується для створення інтерфейсу веб-додатка. Він забезпечує інструменти для створення інтерактивних елементів користувацького інтерфейсу, таких як випадючі списки, графічні компоненти та інші інтерактивні елементи.
2. Dash Cytoscape є бібліотекою для візуалізації мереж у додатках Dash. Вона дозволяє відображати графи та мережі, а також забезпечує інструменти для інтерактивної роботи з ними. Гравці можуть обирати вузли на графі, а граф динамічно оновлюється на основі їх виборів.

3. NetworkX - це бібліотека для створення, маніпулювання та вивчення структур, динаміки і функцій комплексних мереж. Вона використовується для створення графів, визначення вузлів та ребер, а також для розрахунку позицій вузлів у графі.
4. Pandas - це бібліотека для аналізу даних та маніпулювання ними. Вона використовується для створення та обробки таблиць даних, які містять інформацію про вузли та ребра графа, а також для розрахунку виграшів гравців.

3.2.1 Опис основних змінних

Список серверних змінних:

graphGame1 - граф для першої гри

graphGame2 - граф для другої гри

nodes - список вершин для графів ігор

edges - список ребер для графів ігор

player11_decision1 - вибір гравця номер 1 в першій грі

player21_decision1 - вибір гравця номер 2 в першій грі

player12_decision1 - вибір гравця номер 1 в другій грі

player22_decision1 - вибір гравця номер 2 в другій грі

3.3. Опис методу

У програмі розглядається безкоаліційна одночасна гра, в якій два гравці одночасно вибирають місця для своїх поштової відділень і одразу бачать свій виграш. Розрахунок виграшу проводиться наступним чином:

1. Якщо два гравці обралу одну й ту саму вершину, виграш кожного буде дорівнювати $0.25 \cdot n + 0.5$, де n - кількість сусідніх вершин до обраної, а 0.5 - виграш в обраному місті за умовою задачі.
2. Якщо гравці вибрали різні міста, перевіряється, чи є їхні міста сусідніми. Гравці не отримають винагороду за місто, в якому вже розташоване поштове відділення.
3. Після перевірки обраних міст на сусідність, аналізуються міста, сусідні до кожного з обраних. За кожне таке місто гравці отримують по 0.25 .

Рівновага Неша, іншими словами – оптимальне рішення визначається наступним чином:

1. Для кожної гри визначається вершина з найбільшою кількістю сусідів.
2. За припущенням Готелінгу, простір попиту є одновимірним і всі гравці прагнуть розміститися в центрі попиту. Ніщо не завадить їм досягти найбільшого можливого виграшу.
3. Таким чином, очікується, що вершина з найбільшою кількістю сусідів буде рівновагою Неша у цій задачі. Проте, згодом стає зрозуміло, що це не завжди так.

3.4. Приклади роботи програмного продукту

Подивимось на приклади роботи програмного продукту. В програмі представлено 2 графи, які було взяти із книги Еріха Пріснера [9].

У першій грі, початкові графи представлені у виді таких векторів:

$from = c(1, 1, 1, 2, 2, 3, 3, 4, 4, 5, 6)$, $to = c(2, 3, 5, 5, 7, 4, 6, 5, 7, 6, 7)$

Ціль гравців підвищити свій виграш і у разі кооперативної взаємодії вони мають таку можливість. Є аж цілих точки рівноваги Неша: (3,2), (2,3), (7,1) та (1,7) – (рис.3.1. – 3.4)

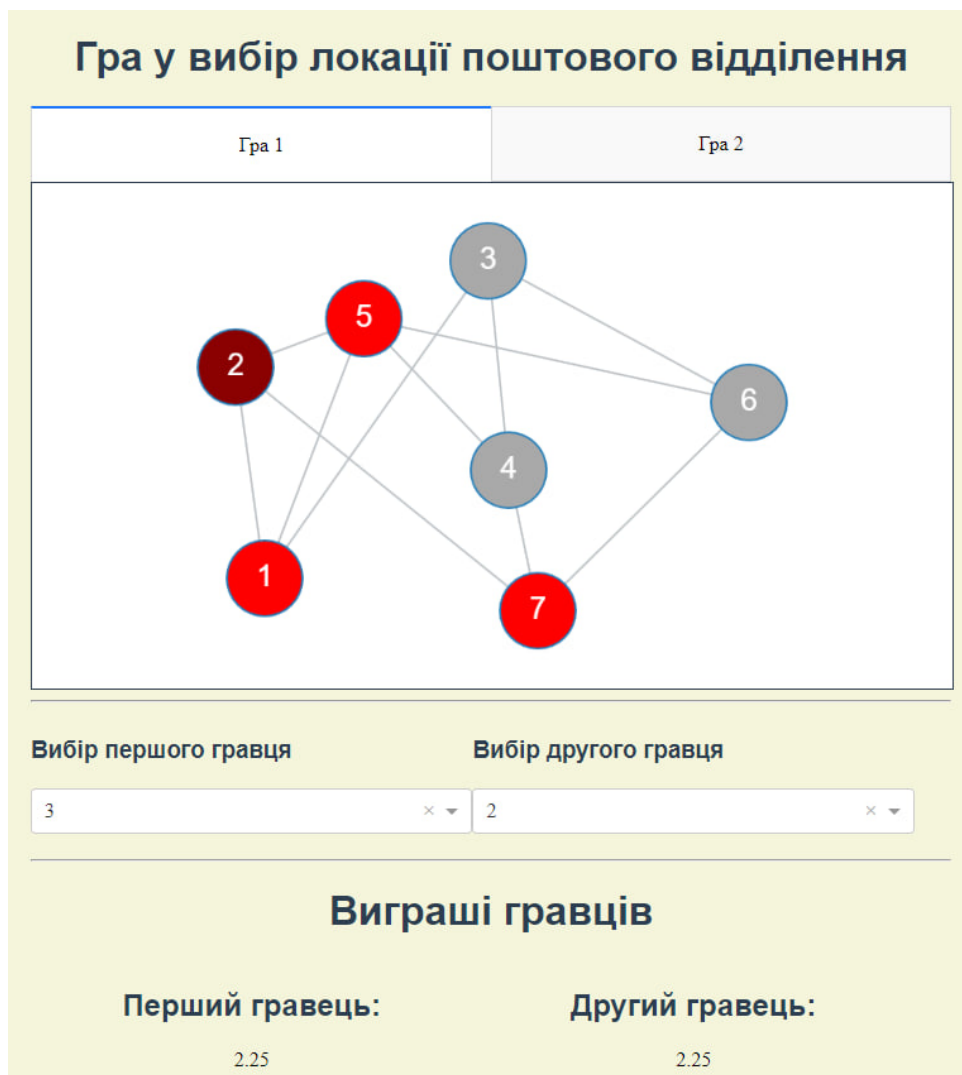


Рисунок 3.1. Розподіли виграшу в точці (3, 2)

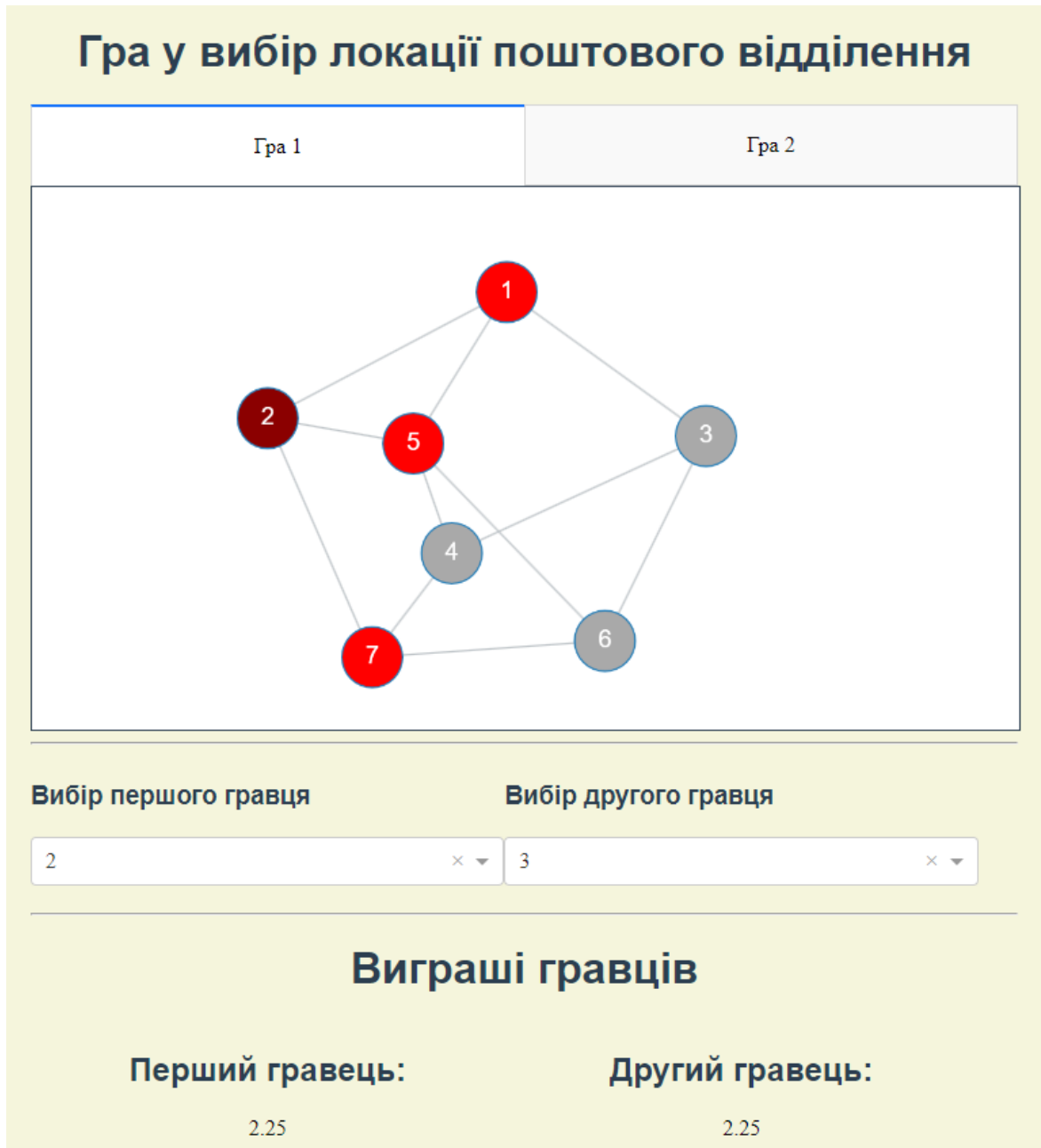


Рисунок 3.2. Розподіли виграшу в точці (2, 3)

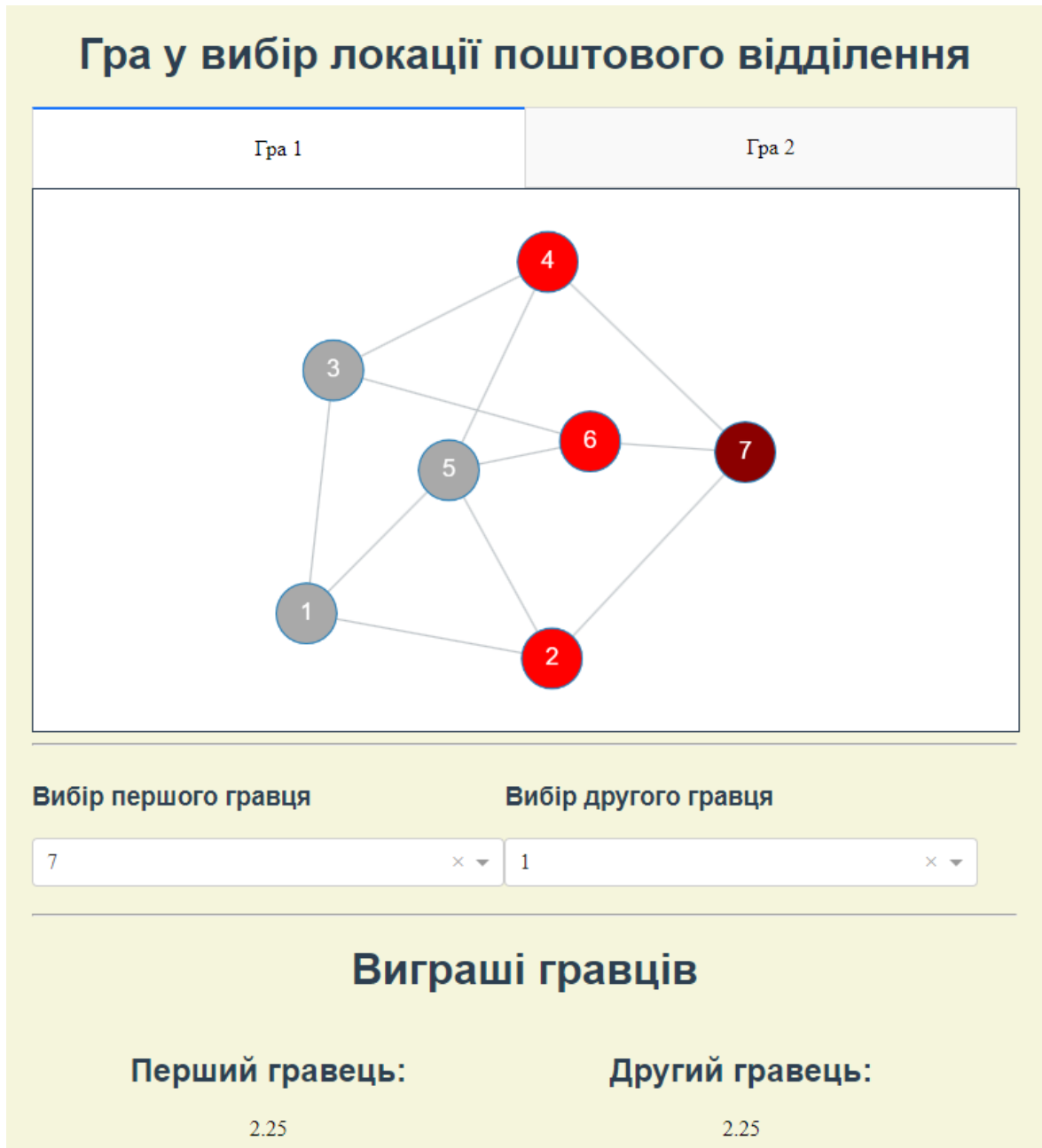


Рисунок 3.3. Розподіли виграшу в точці (7, 1)

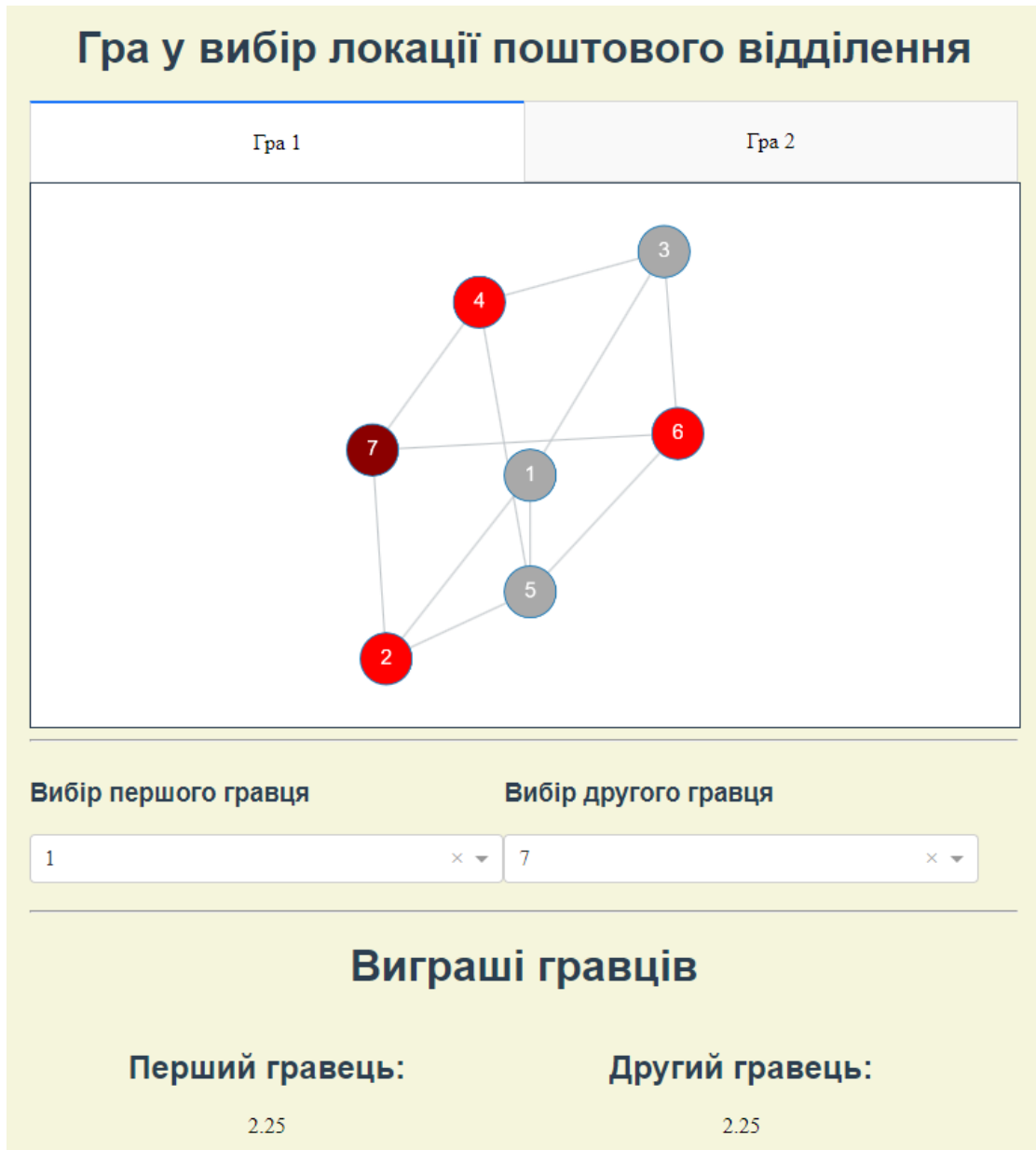


Рисунок 3.4. Розподіли виграшу в точці (1, 7)

Вершина 5 має найбільшу кількість сусідів. За Готелінгом, це буде рівновагою Неша в данному випадку. Тож у випадку некооперативної взаємодії, точка (5,5) буде рівновагою. (рис. 3.5)

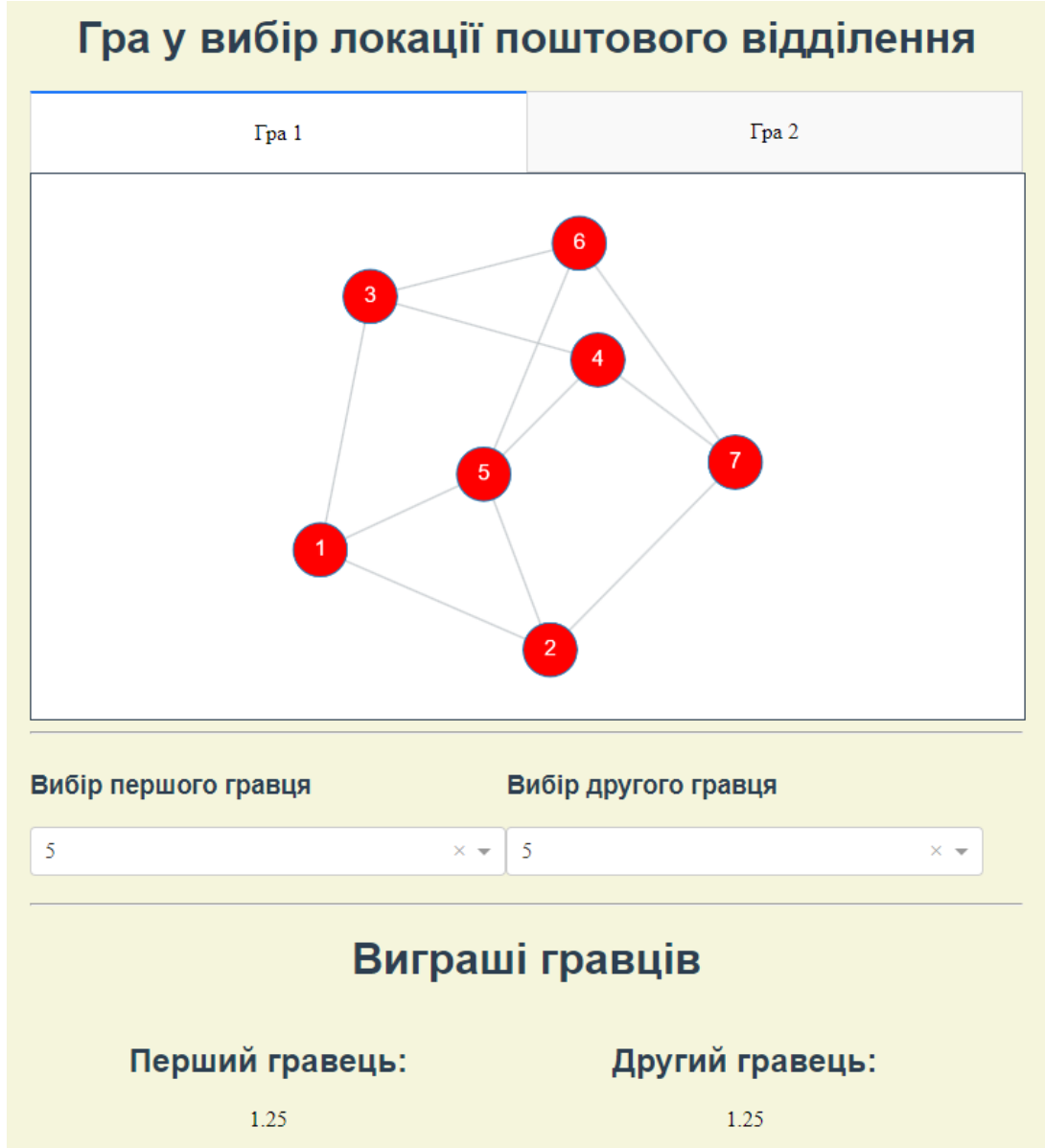


Рисунок 3.5. Розподіли виграшу в точці (5, 5)

Якщо гравці оберуть будь-які однакові вершини, але не вершину 5, то їх максимальний виграш в будь-якому разі буде дорівнювати 1.25 (рис 3.6. - 3.7)

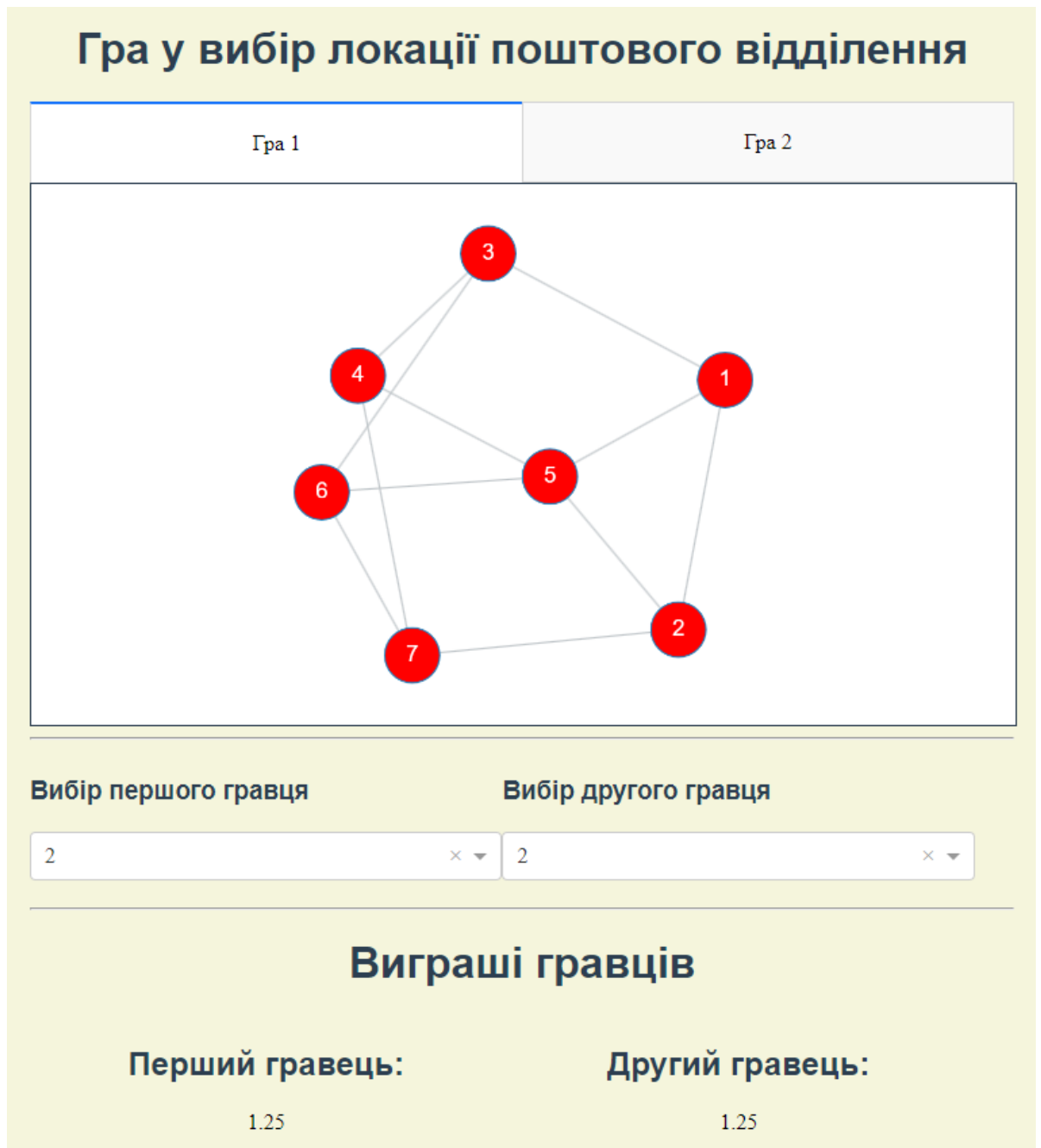


Рисунок 3.6. Розподіли виграшу в точці (2, 2)

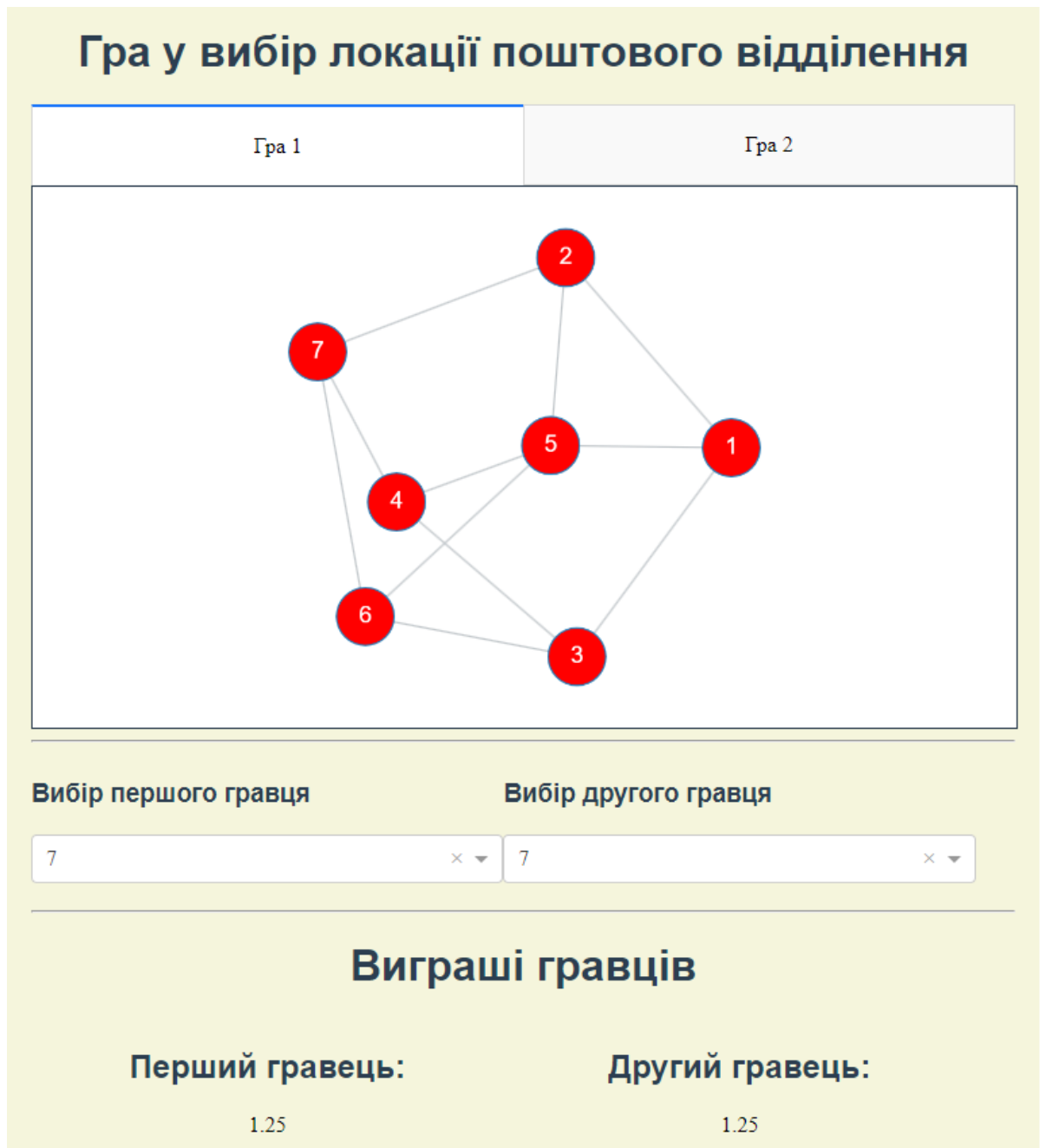


Рисунок 3.7. Розподіли виграшу в точці (7, 7)

У другій грі, початкові графи представлені у виді таких векторів:

$from = c(1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 4, 4, 5, 5, 6, 7, 8)$

$to = c(2, 4, 5, 6, 3, 6, 7, 8, 4, 8, 9, 5, 9, 6, 9, 7, 8, 9)$

У випадку другої гри, гравці отримають найбільший виграш якщо обидва оберуть вершину 2 (точка 2.2) (рис. 3.8)

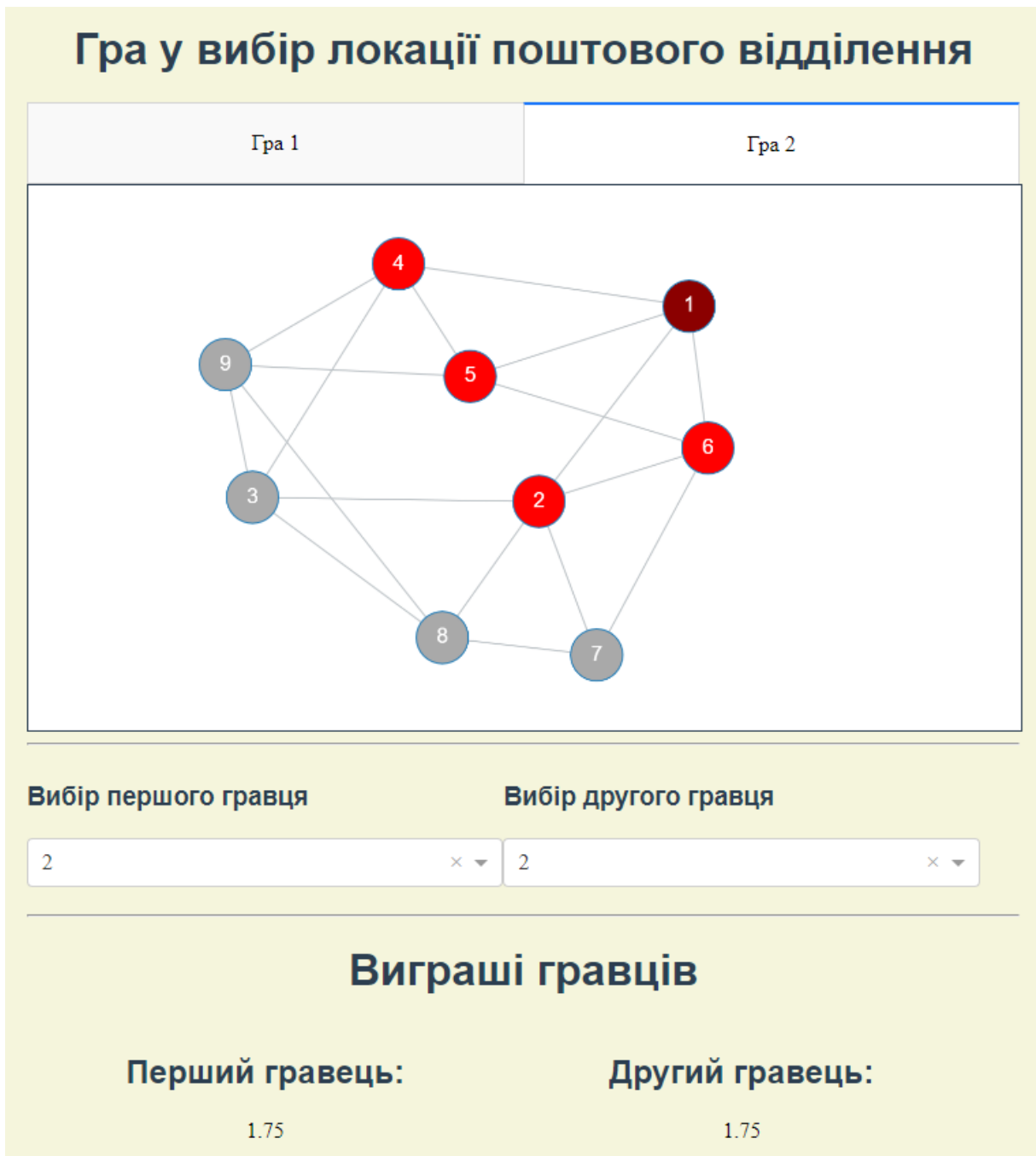


Рисунок 3.8. Розподіли виграшу в точці (2, 2)

Але, навіть не зважаючи на те що вийграш найбільший, це рішення не буде рівновагою Неша, оскільки якщо від нього відхилитись, кожний з гравців може отримати більший виграш. Тобто стратегія кооперації є кращою тут. Наприклад, якщо обрати точки (3;2), (2;3), (8;1), (1;8) - (рис.3.9. – 3.10) . Вони сусідні між собою та в них є одна спільна вершина. Це найбільший можливий виграш, що можна отримати в другій грі.



Рисунок 3.9. Розподіли виграшу в точці (3, 2)

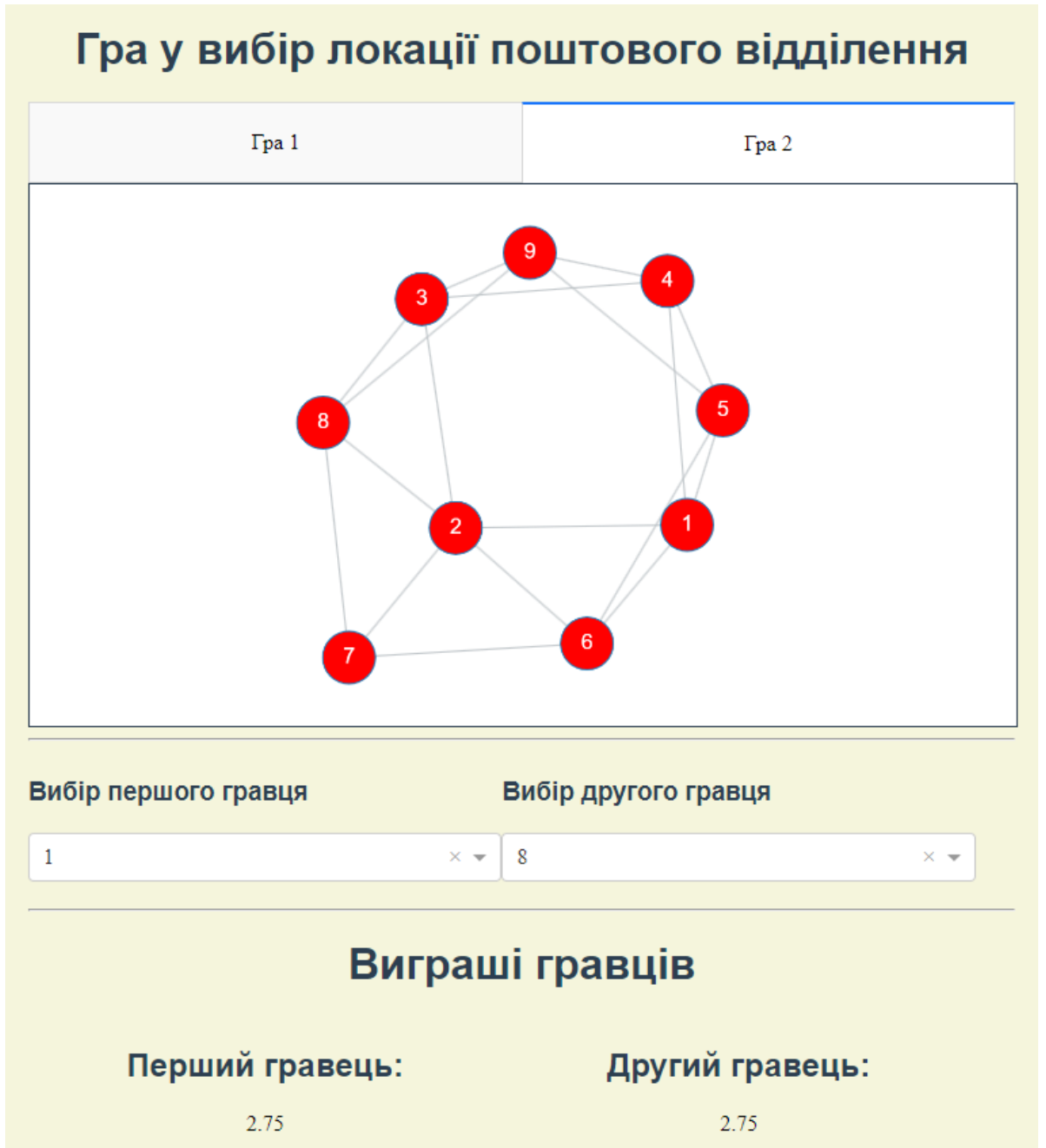


Рисунок 3.10. Розподіли виграшу в точці (1, 8)

3.5.Висновки до розділу 3

В даному розділі ми створили програмний продукт який вирішує базову задачу з вибору локації поштового відділення. Описали використані технології і методологію в цілому. Для реалізації була обрана мова Python, а основними використаними бібліотеками є Dash, Dash Cytoscape, NetworkX та Pandas.

РОЗДІЛ 4 ФУНКЦІОНАЛЬНО-ВАРТІСНИЙ АНАЛІЗ ПРОГРАМНОГО ПРОДУКТУ

4.1. Формування завдання проєктування

Даний розділ призначений для аналізу функціонально-вартісних аспектів застосування теоретико-ігрового підходу у задачі вибору локації для підприємства. Це дозволить оцінити прийняті рішення щодо методів моделювання гри та їхнього впливу на процес вибору оптимальної локації для підприємства. Ці аспекти є критичними для успішного розв'язання завдання та впровадження результатів у практику.

Вибір оптимальної локації для підприємства є складною задачею, яка потребує уважного аналізу різних факторів та варіантів. Етапи аналізу включають в себе вивчення ринкових умов, конкурентної ситуації, аналізу економічної доцільності та ризиків. Неправильний підбір критеріїв аналізу, недостатній обсяг даних або неправильне моделювання впливу різних факторів можуть призвести до неприйнятних результатів.

Цей проєкт спрямований на розробку та застосування теоретико-ігрового підходу у виборі оптимальної локації для підприємства. Основна мета полягає в максимізації прибутковості та мінімізації ризиків у виборі локації. Це може включати моделювання рішень конкурентів та визначення найбільш вигідних стратегій для підприємства.

Економічний аспект лишається важливим у плануванні, оскільки розташування підприємства може суттєво впливати на його прибутковість та конкурентоспроможність. Важливо провести аналіз вартості та користі від кожної альтернативи локації та визначити оптимальне рішення з економічної точки зору.

Аналіз ринку та конкурентів є важливою частиною цього дослідження. У контексті застосування теоретико-ігрового підходу це дослідження має на меті визначення стратегій конкурентів та їх впливу на вибір локації. Це допоможе у розробці ефективних стратегій відповіді та визначенні оптимальної локації для підприємства.

Цей розділ аналізує всі аспекти, які впливають на вибір оптимальної локації для підприємства з використанням теоретико-ігрового підходу. Від вибору критеріїв оцінки до моделювання стратегій конкурентів - всі ці аспекти будуть розглянуті для забезпечення максимальної ефективності та успішності проєкту.

4.2. Обґрунтування функцій програмного продукту

Головна функція F_0 – застосування теоретико-ігрового підходу в задачі вибору локації для підприємства. Програмний продукт повинен забезпечити аналіз різних стратегій гравців, моделювання взаємодії та визначення оптимальної локації для підприємства з урахуванням географічних, економічних та конкурентних факторів. На основі цієї функції були визначені наступні ключові функції програмного продукту:

1. F_1 – збір необхідних даних для аналізу.
2. F_2 – розробка моделей та алгоритмів для визначення взаємодії гравців на ринку.
3. F_3 – імітація та аналіз різних стратегій гравців.
4. F_4 – визначення оптимальної локації на основі результатів аналізу.

Зокрема, розглянуто декілька можливих варіантів реалізації кожної з цих функцій:

Функція F_1 :

- А) Збір географічних даних за допомогою API картографічних сервісів.

Б) Застосування даних з відкритих джерел, таких як бази даних організацій чи географічні системи.

В) Запити до відкритих баз даних про ринкові аналізи та конкурентність обраної локації.

Функція F_2:

А) Розробка моделей ігрової теорії для аналізу взаємодії гравців.

Б) Створення алгоритмів для імітації стратегій конкурентів.

В) Використання аналітичних методів для прогнозування реакції ринку на різні дії.

Функція F_3:

А) Моделювання різних сценаріїв взаємодії гравців на ринку.

Б) Визначення імовірності успішності різних стратегій.

В) Візуалізація результатів моделювання для аналізу та прийняття рішень.

Функція F_4:

А) Оцінка ефективності різних локацій за допомогою математичних моделей.

Б) Побудова геоінформаційних карт для візуалізації оптимальної локації.

В) Аналіз економічної доцільності та ризиків обраної локації для підприємства.

Наступним кроком є створення морфологічної карти, що відображає реалізацію основних функцій програмного продукту. Кінцевий результат буде представлений на рис. 4.1.

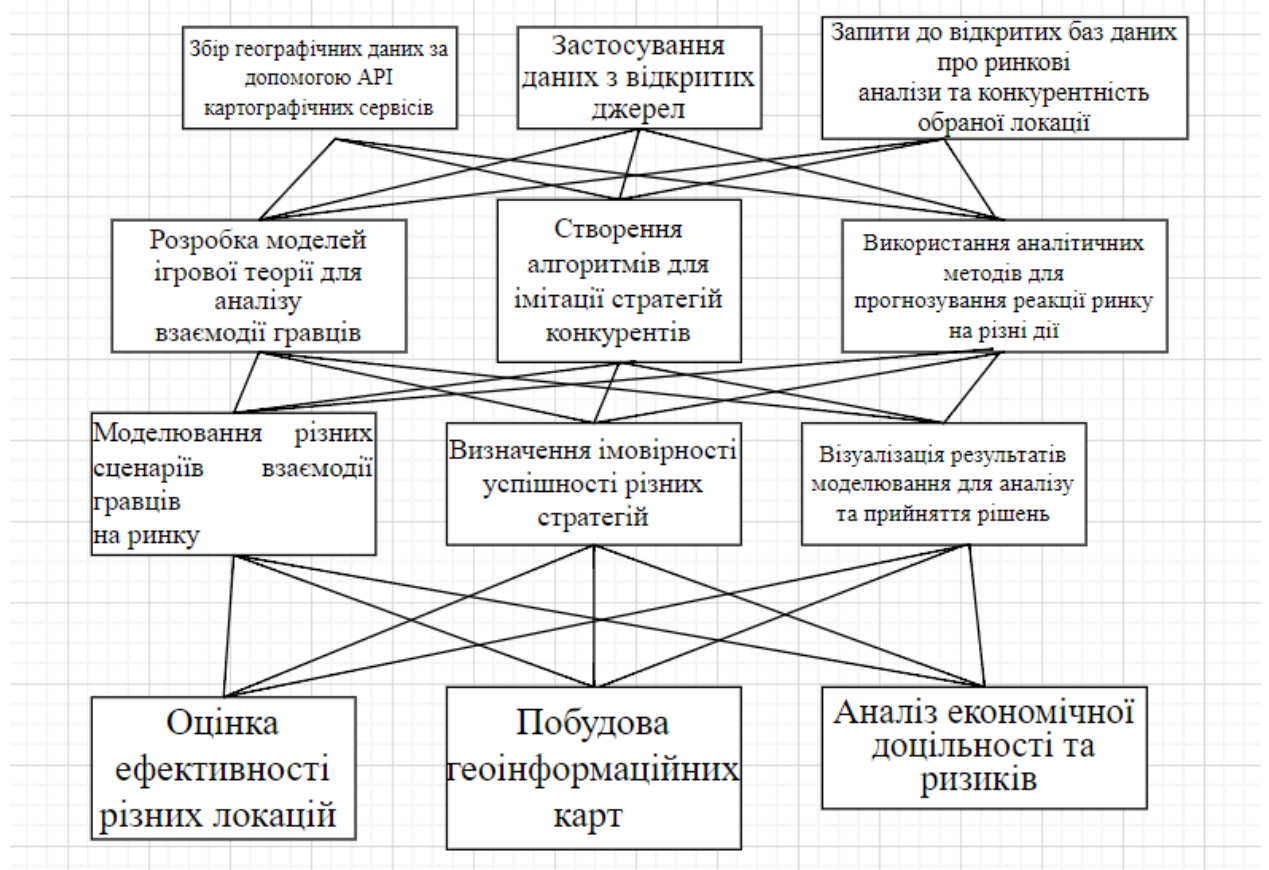


Рис. 4.1 – Морфологічна карта

На основі визначених функцій, представлених варіантів реалізації та сформованої морфологічної карти, було побудовано позитивно-негативну матрицю. Результати показані нижче, у таблиці 4.1.

Таблиця 4.1. Позитивно-негативна матриця

Функції	Переваги	Недоліки
F_1	Збір географічних даних за допомогою API картографічних сервісів. Застосування даних з відкритих джерел. Запити до відкритих баз даних про ринкові аналізи та конкурентність обраної локації.	Переваги: Дозволяє отримати об'єктивні дані про локації. Недоліки: Можливість застарілості даних, обмеженість доступних джерел.

F_2	Розробка моделей ігрової теорії для аналізу взаємодії гравців. Створення алгоритмів для імітації стратегій конкурентів. Використання аналітичних методів для прогнозування реакції ринку на різні дії.	Переваги: Дозволяє зрозуміти взаємодію гравців на ринку, прогнозувати їхні дії. Недоліки: Складність розробки моделей, потреба у великій кількості даних для точних прогнозів.
F_3	Моделювання різних сценаріїв взаємодії гравців на ринку. Визначення імовірності успішності різних стратегій. Візуалізація результатів моделювання для аналізу та прийняття рішень.	Переваги: Дозволяє провести комплексний аналіз сценаріїв та прийняти обґрунтоване рішення. Недоліки: Можливість неточності моделювання, складність інтерпретації результатів.
F_4	Оцінка ефективності різних локацій за допомогою математичних моделей. Побудова геоінформаційних карт для візуалізації оптимальної локації. Аналіз економічної доцільності та ризиків обраної локації для підприємства.	Переваги: Дозволяє об'єктивно оцінити ефективність локацій, візуалізувати дані. Недоліки: Складність розрахунків, потреба у великій кількості даних, можливість недооцінки ризиків.

Після аналізу отриманої позитивно-негативної матриці можна провести фільтрування та вибрати найбільш оптимальні варіанти реалізації для ефективного виконання задач, поставлених для даного проекту.

F_1: Вибір джерел даних з соціальних мереж

Найкращий варіант: Збір географічних даних за допомогою API картографічних сервісів. Цей варіант дозволяє отримати об'єктивні дані про локації, хоча можливість застарілості даних і обмеженість доступних джерел потребує уваги.

F_2: Розробка методів збору даних

Найкращий варіант: Використання аналітичних методів для прогнозування реакції ринку на різні дії. Цей підхід дозволяє зрозуміти взаємодію гравців на ринку та прогнозувати їхні дії, навіть за складних умов.

F_3: Обробка та очищення зібраних даних

Найкращий варіант: Моделювання різних сценаріїв взаємодії гравців на ринку. Це дозволить провести комплексний аналіз сценаріїв та прийняти обґрунтоване рішення, незважаючи на можливість неточності моделювання.

F_4: Аналіз та виявлення трендів та патернів

Найкращий варіант: Оцінка ефективності різних локацій за допомогою математичних моделей. Цей підхід дозволяє об'єктивно оцінити ефективність локацій та візуалізувати дані, навіть за складних умов.

Отже, для ефективного виконання задач можна обрати наступні варіанти реалізації:

F_1: Збір географічних даних за допомогою API картографічних сервісів.

F_2: Використання аналітичних методів для прогнозування реакції ринку на різні дії.

F_3: Моделювання різних сценаріїв взаємодії гравців на ринку.

F_4: Оцінка ефективності різних локацій за допомогою математичних моделей.

4.3. Обґрунтування системи параметрів програмного продукту

Даний розділ передбачає визначення системи параметрів для оцінки та аналізу вибору локації для нашого проекту з використанням теоретико-ігрового підходу. Для цього ми встановимо наступні параметри:

X_1 – Популярність локації серед цільової аудиторії.

X_2 – Економічний потенціал регіону, де знаходиться локація.

X_3 – Рівень конкуренції на ринку в обраній локації.

X_4 – Географічне розташування та доступність транспортних мереж.

X_5 – Потенційні ризики та загрози для бізнесу в обраній локації.

X_6 – Стратегічне значення локації для розвитку бізнесу.

Відповідно до стратегічних цілей та потреб проекту, ми визначили найбільш негативні, середні та найкращі значення параметрів, як показано у таблиці 4.2.

Таблиця 4.2. Основні параметри програмного продукту

Параметр	Умовне позначення	Одиниця виміру	Значення
Об'єм оперативної пам'яті для роботи системи	X_1	Мб	4096
Об'єм написаного програмного коду	X_2	Кількість рядків	25000
Об'єм задіяних ресурсів центрального процесору	X_3	Відсотки	70
Об'єм затраченого часу на обробку запитів	X_4	Мілісекунди	750
Точність визначення трендів	X_5	Відсотки	80

Кількість підтримуваних соціальних медіа-платформ	X_6	Кількість платформ	2
Масштабованість системи	X_7	Рівень	Високий

Після визначення значень параметрів можна розпочати аналіз коефіцієнта технічного рівня. Враховуючи вимоги та реальні умови експлуатації системи, ми можемо оцінити її продуктивність, ефективність та відповідність потребам ринку.

Кожен з параметрів є критичним для забезпечення стабільної роботи та високої продуктивності системи аналізу соціальних медіа-даних. Наприклад, об'єм оперативної пам'яті (X_1) впливає на можливість обробки великої кількості даних в реальному часі, а об'єм написаного програмного коду (X_2) відображає складність та функціональність системи. Об'єм задіяних ресурсів центрального процесору (X_3) та час на обробку запитів (X_4) є важливими показниками ефективності системи.

Точність визначення трендів (X_5) та кількість підтримуваних платформ (X_6) визначають можливості системи в аналізі даних з різних джерел та коректність отриманих результатів. Масштабованість системи (X_7) впливає на можливість розширення функціоналу та адаптацію до зростання обсягів даних.

Оцінка цих параметрів дозволяє зробити висновки щодо відповідності розробленої системи сучасним вимогам та її конкурентоспроможності на ринку.

4.4. Аналіз експертного оцінювання параметрів

Четвертий етап функціонально-вартісного аналізу програмного продукту включає визначення вагомості кожного параметра в загальній кількості розглянутих під час оцінювання параметрів за допомогою методу попарного порівняння.

Для цього спочатку було знайдено коефіцієнти вагомості, визначено ступінь важливості параметрів шляхом присвоєння їм різних рангів експертами-фахівцями в галузі. Результати продемонстровано в табл. 4.3.

Таблиця 4.3. Результати ранжування параметрів

Позначення параметра	Одиниці виміру	Ранг параметра за оцінкою експерта
X_1	Мб	3
X_2	Кількість рядків	1
X_3	Відсотки	4
X_4	Мілісекунди	2
X_5	Відсотки	4
X_6	Кількість платформ	2
X_7	Рівень	1

Після чого було обчислено параметри:

Загальна сума рангів (4.1):

$$R_i = \sum_{j=1}^N r_{ij} = \frac{Nn(n+1)}{2} = \frac{7 \cdot 7 \cdot (7+1)}{2} = 196 \quad (4.1)$$

2. Середня сума рангів (4.2):

$$T = \frac{1}{n} R_i = 28 \quad (4.2)$$

3. Відхилення суми рангів параметрів від середньої суми рангів (4.3):

$$\Delta_i = R_i - T \quad (4.3)$$

4. Загальна сума квадратів відхилення (4.4):

$$S = \sum_{i=1}^N \Delta_i^2 = 182 \quad (4.4)$$

5. Коефіцієнт узгодженості (4.5):

$$W = \frac{12S}{N^2(n^3-n)} = \frac{12 \cdot 182}{7^2 \cdot (7^3-7)} \approx 0.54 \quad (4.5)$$

Для оцінки узгодженості між поглядами експертів було розглянуто нерівність (сформовану на базі значення нормативного коефіцієнта для параметрів програмних продуктів $W_{\text{норм}} = 0.60$) наступного виду $W > W_{\text{норм}}$, що свідчить про достатньо високий рівень узгодженості. Оскільки результат виявився позитивним, наступним кроком було виконано попарне порівняння параметрів, що продемонстровано в табл. 4.4.

Таблиця 4.4. Попарне порівняння параметрів

Параметри	Експерти	Кінцева оцінка	Числове значення
	1	2	3
X_1 і X_2	>	>	>
X_1 і X_3	<	>	<
X_1 і X_4	>	>	>
X_1 і X_5	<	>	<
X_1 і X_6	>	>	>
X_1 і X_7	>	>	>

X ₂ і X ₃	<	<	<
X ₂ і X ₄	<	<	<
X ₂ і X ₅	<	<	<
X ₂ і X ₆	<	<	<
X ₂ і X ₇	<	<	<
X ₃ і X ₄	>	>	>
X ₃ і X ₅	>	>	>
X ₃ і X ₆	>	>	>
X ₃ і X ₇	>	>	>
X ₄ і X ₅	<	<	<
X ₄ і X ₆	<	<	<
X ₄ і X ₇	<	<	<
X ₅ і X ₆	>	>	>
X ₅ і X ₇	>	>	>
X ₆ і X ₇	>	>	>

Далі за результатами числових значень коефіцієнтів переваг на основі підсумкових оцінок, було складено квадратну матрицю $A = \|x_{ij}\|$ та розраховано пріоритетність K_{Bi} для кожного із параметрів доки різниця між ітераціями розрахунку даних вагомості не стала меншими за 5%.

Розрахунки для першої ітерації відбувались за формулами 4.6 і 4.7:

$$K_{Bi} = \frac{b_i}{\sum_{i=1}^n b_i} \quad (4.6)$$

$$b_i = \sum_{i=1}^N a_{ij} \quad (4.7)$$

Для поточних ітерацій використовувались 4.8 і 4.9:

$$K_{Bi} = \frac{b_i'}{\sum_{i=1}^n b_i'} \quad (4.8)$$

$$b_i' = \sum_{i=1}^N a_{ij} b_j \quad (4.9)$$

Результати наведено в табл. 4.5:

Табл. 4.5. Розрахунок вагомості параметрів

Параметри x_i	Параметри x_j	Результат порівняння
	X1	X2
X1	1.0	1.2
X2	1.2	1.0
X3	0.8	0.4
X5	1.0	0.6
X6	0.9	0.7
X7	1.1	0.9

Оскільки різниця між коефіцієнтами вагомості параметрів вже на другій ітерації почала дорівнювати 2% (0.02), що вже є меншим від 5% (0.05), то було призупинено подальші розрахунки і визначено дану таблицю як остаточну.

4.5. Аналіз рівня якості варіантів реалізації функцій

На основі табл. 4.1, було оцінено потенційні абсолютні значення параметрів для нашого проєкту. Кінцевим результатом оцінки виступає табл. 4.6., де проведено розрахунки показників технічного рівня, що враховує варіанти реалізації функцій та параметри на основі раніше обчисленого коефіцієнта.

Табл. 4.6. Розрахунок показників рівня якості

Осно вні функці ї	Варі ант реаліза ції функці ї	Парам етри здіяні у реалізації	Абсол ютне значення параметр а	Баль на оцінка параме тра	Коефіц ієнт вагомості параметр а	Коефіц ієнт рівня якості
F_1	A	X_3	Високи й	8	0.25	2.00
	B	X_3	Високи й	8	0.25	2.00
	C	X_3	Високи й	8	0.25	2.00
F_2	A	X_5	Середні й	6	0.30	1.80
	B	X_5	Середні й	6	0.30	1.80
	C	X_5	Середні й	6	0.30	1.80
F_3	A	X_8	Низьки й	4	0.20	0.80
	B	X_8	Низьки й	4	0.20	0.80
	C	X_8	Низьки й	4	0.20	0.80
F_4	A	X_2	Високи й	9	0.25	2.25
	B	X_2	Високи й	9	0.25	2.25

	С	X_2	Високи й	9	0.25	2.25
--	---	-----	-------------	---	------	------

Після просумування показників технічного рівня для обох варіантів реалізації системи (А та В) ми отримали наступні узагальнені коефіцієнти рівня якості:

1. Коефіцієнт рівня якості для варіанту А: $2.00 + 1.80 + 0.80 + 2.25 = 6.85$
2. Коефіцієнт рівня якості для варіанту В: $2.00 + 1.80 + 0.80 + 2.25 = 6.85$

Отже, при порівнянні обох варіантів реалізації програмного продукту, функціональний аналіз дозволяє визначити, що обидва варіанти (А та В) мають однаковий рівень якості, оскільки їх значення рівня якості однакові.

4.6. Економічний аналіз варіантів розробки ПП

Для визначення загальної трудомісткості системи даного проєкту було використано формулу 4.6.1:

$$T_0 = T_P \cdot K_P \cdot K_{СК} \cdot K_M \cdot K_{СТ} \cdot K_{СТМ} \quad (4.10)$$

Після чого визначено початкові дані системи завдання (розробка програмного продукту) з метою отримання даних про поправочні коефіцієнти для розрахунку загальної трудомісткості. Даний перелік включає в себе:

- Ступінь новизни групи задач – «В» (задачі, які передбачають використання нових методів оцінки локацій).
- Складність алгоритму – «2» (алгоритми аналізу та порівняння локацій).
- Складність організації контролю вхідної та вихідної інформації – «3.2 + 3.6» (вихідні дані та документи середнього розміру і структури).
- Вид використовуючої інформації – «Географічна база даних».

□ Мова програмування – «R».

□ Використання стандартних модулів програмування. З даного переліку було сформовано вже поправочні коефіцієнти за таблицями умов із наданих методичних вказівок:

1. $T_p - 150$
2. $K_{\Pi} - 2.50$
3. $K_{СК} - 1.30$
4. $K_M - 1.15$
5. $K_{СТ} - 0.80$
6. $K_{СТМ} - 1.10$

Таким чином в наявності опинились всі елементи необхідні для складання початкового рівняння (4.10):

$$T_0 = 150 \cdot 2.50 \cdot 1.30 \cdot 1.15 \cdot 0.80 \cdot 1.10 \approx 397.13 \text{ (людино днів)}$$

Розрахувавши загальну трудомісткість програмного продукту, її також було розраховано для двох варіантів реалізації:

$$T_1 = 397.13 \cdot 4 \cdot 7 \approx 11119.64 \text{ (людино днів)} \quad (4.11)$$

$$T_2 = 397.13 \cdot 4 \cdot 7 \approx 11119.64 \text{ (людино днів)}$$

Результат розрахунків виявився доволі логічним, оскільки хоч підхід та функціонал в обох бібліотеках до створенню графічного інтерфейсу може відрізнитись, сама його структура з елементами повинна лишатись однаковою.

Оскільки у розробці бере участь один науковець у сфері даних з окладом 60000 грн, то середня заробітня плата складає (4.12):

$$C_{\text{ч}} = \frac{M}{T_m \cdot t} = \frac{70000}{3 \cdot 21 \cdot 8} = 138.89 \text{ (грн)}$$

Також було обчислено зарплату для обох варіантів реалізації (4.13):

$$C_{зп} = C_{ч} \cdot T_i \cdot K_d = 138.89 \cdot 11119.64 \cdot 1.1 \approx 1,701,234.85 \text{ (грн)} \quad (4.13)$$

Відраховано соціальний внесок від заробітньої плати (4.14):

$$C_{від} = C_{зп} \cdot 0.22 = 1,701,234.85 \cdot 0.22 \approx 374,271.67 \text{ (грн)} \quad (4.14)$$

Для знаходження вартості однієї машино-години, спочатку було розраховано перелік інших змінних.

Машинний час (4.15), враховуючи додаткову заробітню плату і відрахування соціального внеску:

$$C_r = 12 \cdot M \cdot K_3 = 12 \cdot 60,000 \cdot 0.25 = 180,000 \text{ (грн)} \quad (4.15)$$

$$C_{зп} = C_r \cdot (1 + K_3) = 60,000 \cdot 0.25 = 180,000 \text{ (грн)}$$

$$C_{від} = C_{зп} \cdot 0.22 = 225,000 \cdot 0.22 = 49,500 \text{ (грн)}$$

Амортизаційні відрахування при вартості обладнання в 100,000 грн та амортизації в 20% (4.16)

$$C_A = K_{TM} \cdot K_A \cdot C_{ПР} = 1.10 \cdot 0.25 \cdot 100,000 = 27,500 \text{ (грн)} \quad (4.16)$$

Вартість ремонту та профілактики (4.17):

$$C_P = K_{TM} \cdot K_P \cdot C_{ПР} = 1.10 \cdot 0.05 \cdot 100,000 = 5,500 \text{ (грн)} \quad (4.17)$$

Ефективний годинний фонд часу ПК за рік (4.17):

$$\begin{aligned} T_{ЕФ} &= (D_K - D_B - D_C - D_P) \cdot t_3 \cdot K_B = \\ &= (365 - 104 - 14 - 10) \cdot 8 \cdot 0.7 = 1663.2 \text{ (год.)} \end{aligned} \quad (4.17)$$

Кількість коштів витрачених на оплату електроенергії (4.18):

$$C_{\text{ЕЛ}} = T_{\text{ЕФ}} \cdot N_C \cdot K_3 \cdot C_{\text{ен}} \cdot K_B = 1663.2 \cdot 0.15 \cdot 0.15 \cdot 6.00 = 224.55 \text{ (грн.)} \quad (4.18)$$

Накладні витрати (4.19):

$$C_H = C_{\text{ПР}} \cdot 0.65 = 100,000 \cdot 0.65 = 65,000 \text{ (грн.)} \quad (4.19)$$

Вартість експлуатації (4.20):

$$C_{\text{ЕКС}} = C_{\text{ЗП}} + C_{\text{ВІД}} + C_A + C_P + C_{\text{ЕЛ}} + C_H = 225,000 + 49,500 + 27,500 + 5,500 + 224.55 + 65,000 \approx 372,724.55 \text{ (грн.)} \quad (4.20)$$

Вартість однієї машино-години (4.21):

$$C_{\text{М-Г}} = \frac{C_{\text{ЕКС}}}{T_{\text{ЕФ}}} = \frac{372,724.55}{1663.2} = 224.100 \text{ (грн.)} \quad (4.21)$$

Для обох варіантів реалізації (4.22):

$$C_M = C_{\text{М-Г}} \cdot T = 224.100 \cdot 11119.64 \approx 2,491,920.88 \text{ (грн.)} \quad (4.22)$$

Накладні витрати (4.23):

$$C_H = C_{\text{ЗП}} \cdot 0.65 = 225,000 \cdot 0.65 = 146,250 \text{ (грн.)} \quad (4.23)$$

Загальна вартість розробки програмного продукту за обома варіантами становила однакове значення, оскільки обидві варіанти не потребують особливих витрат на якісь підписки чи додаткові типи послуг (4.24):

$$C_{\text{ПП}} = C_{\text{ЗП}} + C_{\text{ВІД}} + C_{\text{М}} + C_{\text{Н}} = 225,000 + 49,500 + 2,491,920.88 + 146,250 \approx 2,912,670.88 \text{ (грн.)} \quad (4.24)$$

4.7. Вибір кращого варіанта техніко-економічного рівня

Останнім етапом було обрано найкращий варіант реалізації. Для цього спочатку розраховано коефіцієнт техніко-економічного рівня для двох варіантів реалізації відповідно (4.25):

$$K_{\text{ТЕР } j} = \frac{K_{\text{К } j}}{C_{\text{ПП } j}} = \frac{8.45}{2,500,000} = 3.38 \cdot 10^{-6} \quad (4.25)$$

$$K_{\text{ТЕР } j} = \frac{K_{\text{К } j}}{C_{\text{ПП } j}} = \frac{7.25}{2,500,000} = 2.90 \cdot 10^{-6}$$

Отже, оскільки величина коефіцієнта техніко-економічного рівня визначає його ефективність, то було прийнято рішення імплементувати перший варіант, який налічує в собі наступні опції:

- Мова програмування – Python;
- Реалізація важливої постановки з допомогою вбудованих функцій;
- Використання стандартного інтерфейсу для побудови значень.

4.8. Висновки до 4 розділу

В результаті проведеного функціонально-вартісного аналізу програмного продукту, призначеного для аналізу соціальних медіа-даних з метою виявлення трендів і визначення потреб ринку, було отримано висновки щодо ефективності обраних технологій та економічних стратегій для даного продукту. Оцінка основних характеристик програмного забезпечення виявила різні варіанти реалізації, які забезпечують оптимальну стратегію вибору для досягнення мети аналізу соціальних медіа-даних.

Метод функціонально-вартісного аналізу використовувався для об'єктивної оцінки реальної вартості продукту та виявлення можливостей зниження витрат. Це сприятиме подальшому потенційному впровадженню продукту в реальні умови та розвитку спеціалістів у галузі аналізу соціальних медіа-даних, що є важливим аспектом у розвитку сучасних технологій і маркетингових стратегій.

ВИСНОВКИ

В даній роботі було досліджено контекст виникнення задач з вибору локацій як таких, персоналії людей які зробили внесок їх розвиток та практичні приклади їх постановок та вирішення в виді FLP, TPLP, MTPLP, FTPLP.

Описані основні принципи теорії ігор, в основі яких лежать парадигми кооперативної і некооперативної взаємодії гравців (учасників гри).

Створено програмний продукт, який дозволяє ознайомитись із теоретико-ігровим підходом до задачі вибору локації шлях розв'язання задачі про розміщення двох поштових відділень.

Описані потенційні витрати на реалізацію проекту в сукупності.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Giuseppe Bruno, Andrea Genovese & Gennaro Improta (2014) A historical perspective on location problems, BSHM Bulletin: Journal of the British Society for the History of Mathematics, 29:2, 83-97
URL: <https://www.tandfonline.com/doi/abs/10.1080/17498430.2014.889266>
2. Brimberg, J., Wesolowsky, G. Minisum (2002) Location with Closest Euclidean Distances. Annals of Operations Research 111, 151–165
URL: <https://doi.org/10.1023/A:1020901719463>
3. Berman, O., Drezner, Z. and Wesolowsky, G.O. (2005), The facility and transfer points location problem. International Transactions in Operational Research, 12: 387-402.
URL: <https://doi.org/10.1111/j.1475-3995.2005.00514.x>
4. Nadine Schuurman, Nathaniel J Bell, Randy L'Heureux & Syed M Hameed (2009) Modelling optimal location for pre-hospital helicopter emergency medical services, BMC Emergency Medicine
URL: <https://bmcemergmed.biomedcentral.com/articles/10.1186/1471-227X-9-6>
5. Liang Zhou ,Shaohua Wang, Zhibang Xu (2020) A Multi-factor Spatial Optimization Approach for Emergency Medical Facilities in Beijing, GIS in Healthcare
URL: <https://www.mdpi.com/2220-9964/9/6/361>
6. REopt: Renewable Energy Integration & Optimization
URL: <https://www.nrel.gov/reopt/>

7. Yongji Cao, Yi Zhang, Hengxu Zhang, Pengfei Zhang (2019) Complementarity assessment of wind-solar energy sources in Shandong province based on NASA, The Journal of Engineering Volume 2019, Issue 18
URL: <https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/joe.2018.9367>
8. Brown S. Harold Hotelling and the principle of minimum differentiation. Progress in Human Geography. 1989;13(4):471-493.
URL: <https://doi.org/10.1177/030913258901300401>
9. Prisner, Erich. Game Theory Through Examples. 1st ed., Mathematical Association of America, 2014. JSTOR,
URL: <http://www.jstor.org/stable/10.4169/j.ctt6wpwgj>.
10. Бреженко Н. С. Теоретико-ігровий підхід в задачі вибору локації: дипломна робота ... бакалавра: 124 Системний аналіз. Київ, КПІ ім. І. Сікорського, 2021.
URL: <https://ela.kpi.ua/handle/123456789/43898>
11. Бреженко Н. С. Теоретико-ігровий підхід в задачі вибору локації: магістерська дис. : 124 Системний аналіз. Київ, КПІ ім. І. Сікорського, 2022.
URL: <https://ela.kpi.ua/handle/123456789/53394>
12. Jean-François Caulier, Alexandre Skoda, Emily Tanimura (2017) Allocation Rules for Networks Inspired by Cooperative Game-Theory
URL: <https://www.cairn.info/revue-d-economie-politique-2017-4-page-517.htm>
13. Aaron McDonald (2005) An Introduction to Non-Cooperative Game Theory, University of Utah Math Circle.
URL: <https://www.math.utah.edu/mathcircle/notes/amcdonall.pdf>

ДОДАТОК А. ТЕКСТ ПРОГРАМИ

```
import dash
from dash import dcc, html
from dash.dependencies import Input, Output
import dash_cytoscape as cyto
import networkx as nx

app = dash.Dash(__name__)

app.layout = html.Div([
    html.H1("Гра у вибір локації поштового відділення", style={
        'textAlign': 'center',
        'font-family': 'Arial, sans-serif',
        'color': '#2C3E50',
        'margin-bottom': '20px'
    }),
    dcc.Tabs([
        dcc.Tab(label='Гра 1', children=[
            cyto.Cytoscape(
                id='playingGraph1',
                style={'width': '100%', 'height': '400px', 'border': '1px solid #2C3E50',
'background-color': '#FFFFFF'},
                elements=[],
                layout={'name': 'preset'},
                stylesheet=[{
                    'selector': 'node',
                    'style': {
                        'content': 'data(label)',
```

```
        'text-valign': 'center',
        'text-halign': 'center',
        'width': '75px',
        'height': '75px',
        'font-size': '30px',
        'background-color': '#A9A9A9',
        'color': '#fff',
        'border-width': '2px',
        'border-color': '#2980B9'
    }
}, {
    'selector': 'node.selected',
    'style': {
        'background-color': '#FF0000'
    }
}, {
    'selector': 'node.highlight',
    'style': {
        'background-color': '#8B0000'
    }
}, {
    'selector': 'edge',
    'style': {
        'line-color': '#BDC3C7',
        'width': '2px'
    }
}]
),
html.Hr(),
html.Div([
```

```

    html.Div([
        html.H3('Вибір першого гравця', style={'font-family': 'Arial, sans-
        serif', 'color': '#2C3E50'}),
        dcc.Dropdown(id='choice1Player1', options=[{'label': str(i), 'value':
        i} for i in range(1, 8)], value=None)
    ], style={'width': '48%', 'display': 'inline-block'}),
    html.Div([
        html.H3('Вибір другого гравця', style={'font-family': 'Arial, sans-
        serif', 'color': '#2C3E50'}),
        dcc.Dropdown(id='choice1Player2', options=[{'label': str(i), 'value':
        i} for i in range(1, 8)], value=None)
    ], style={'width': '48%', 'display': 'inline-block'}),
    ], style={'margin-bottom': '20px'}),
    html.Hr(),
    html.H1("Виграші гравців", style={'textAlign': 'center', 'font-family':
    'Arial, sans-serif', 'color': '#2C3E50'}),
    html.Div([
        html.Div([html.H2("Перший гравець:", style={'font-family': 'Arial,
        sans-serif', 'color': '#2C3E50'}), html.Div(id='player1')], style={'width': '48%',
        'display': 'inline-block', 'textAlign': 'center'}),
        html.Div([html.H2("Другий гравець:", style={'font-family': 'Arial,
        sans-serif', 'color': '#2C3E50'}), html.Div(id='player2')], style={'width': '48%',
        'display': 'inline-block', 'textAlign': 'center'}),
    ], style={'margin-bottom': '20px'})
]),
    dcc.Tab(label='Гра 2', children=[
        cyto.Cytoscape(
            id='playingGraph2',
            style={'width': '100%', 'height': '400px', 'border': '1px solid #2C3E50',
            'background-color': '#FFFFFF'},

```

```
elements=[],
layout={'name': 'preset'},
stylesheet=[{
  'selector': 'node',
  'style': {
    'content': 'data(label)',
    'text-valign': 'center',
    'text-halign': 'center',
    'width': '75px',
    'height': '75px',
    'font-size': '30px',
    'background-color': '#A9A9A9',
    'color': '#fff',
    'border-width': '2px',
    'border-color': '#2980B9'
  }
}, {
  'selector': 'node.selected',
  'style': {
    'background-color': '#FF0000'
  }
}, {
  'selector': 'node.highlight',
  'style': {
    'background-color': '#8B0000'
  }
}, {
  'selector': 'edge',
  'style': {
    'line-color': '#BDC3C7',
```

```

        'width': '2px'
    }
}
}]
),
html.Hr(),
html.Div([
    html.Div([
        html.H3('Вибір першого гравця', style={'font-family': 'Arial, sans-
serif', 'color': '#2C3E50'}),
        dcc.Dropout(id='choice1Player12', options=[{'label': str(i), 'value':
i} for i in range(1, 10)], value=None)
    ], style={'width': '48%', 'display': 'inline-block'}),
    html.Div([
        html.H3('Вибір другого гравця', style={'font-family': 'Arial, sans-
serif', 'color': '#2C3E50'}),
        dcc.Dropout(id='choice1Player22', options=[{'label': str(i), 'value':
i} for i in range(1, 10)], value=None)
    ], style={'width': '48%', 'display': 'inline-block'}),
    ], style={'margin-bottom': '20px'}),
html.Hr(),
html.H1("Виграші гравців", style={'textAlign': 'center', 'font-family':
'Arial, sans-serif', 'color': '#2C3E50'}),
html.Div([
    html.Div([html.H2("Перший гравець:", style={'font-family': 'Arial,
sans-serif', 'color': '#2C3E50'}), html.Div(id='player21')], style={'width': '48%',
'display': 'inline-block', 'textAlign': 'center'}),
    html.Div([html.H2("Другий гравець:", style={'font-family': 'Arial,
sans-serif', 'color': '#2C3E50'}), html.Div(id='player22')], style={'width': '48%',
'display': 'inline-block', 'textAlign': 'center'}),
    ], style={'margin-bottom': '20px'})

```

```

    ]),
    ])
], style={'background-color': '#F5F5DC', 'padding': '20px'})

```

```
def generate_elements(nodes, edges, choice1=None, choice2=None, game=1):
```

```
    G = nx.Graph()
```

```
    G.add_nodes_from(nodes)
```

```
    G.add_edges_from(edges)
```

```
    pos = nx.spring_layout(G)
```

```
    elements = []
```

```
    for node in nodes:
```

```
        x, y = pos[node]
```

```
        node_class = "
```

```
        if choice1 == choice2 and (game != 2 and choice1 != 2):
```

```
            node_class = 'selected'
```

```
        elif game == 1:
```

```
            if (choice1, choice2) in [(1, 7), (7, 1)] and node in [7, 4, 6, 2]:
```

```
                node_class = 'highlight' if node == 7 else 'selected'
```

```
            elif (choice1, choice2) in [(2, 3), (3, 2)] and node in [2, 1, 5, 7]:
```

```
                node_class = 'highlight' if node == 2 else 'selected'
```

```
        elif game == 2:
```

```
            if (choice1, choice2) == (2, 2):
```

```
                if node == 1:
```

```
                    node_class = 'highlight'
```

```
                elif node in [2, 4, 5, 6]:
```

```
                    node_class = 'selected'
```

```
            elif (choice1, choice2) == (3, 2) or (choice1, choice2) == (1, 8):
```

```
                node_class = 'selected'
```

```
            elif (choice1, choice2) == (9, 9) and node in [8, 1, 4, 5, 6, 7]:
```

```

        node_class = 'highlight' if node == 8 else 'selected'
    elements.append({
        'data': {'id': str(node), 'label': str(node)},
        'position': {'x': x * 300, 'y': y * 300},
        'classes': node_class
    })

for edge in edges:
    elements.append({'data': {'source': str(edge[0]), 'target': str(edge[1])}})

return elements

@app.callback(
    Output('playingGraph1', 'elements'),
    [Input('choice1Player11', 'value'), Input('choice1Player21', 'value')]
)
def update_graph1(choice1, choice2):
    nodes = list(range(1, 8))
    edges = [(1, 2), (1, 3), (1, 5), (2, 5), (2, 7), (3, 4), (3, 6), (4, 5), (4, 7), (5, 6), (6,
7)]
    return generate_elements(nodes, edges, choice1, choice2, game=1)

@app.callback(
    Output('playingGraph2', 'elements'),
    [Input('choice1Player12', 'value'), Input('choice1Player22', 'value')]
)
def update_graph2(choice1, choice2):
    nodes = list(range(1, 10))
    edges = [(1, 2), (1, 4), (1, 5), (1, 6), (2, 3), (2, 6), (2, 7), (2, 8), (3, 4), (3, 8), (3,
9), (4, 5), (4, 9), (5, 6), (5, 9), (6, 7), (7, 8), (8, 9)]

```

```
return generate_elements(nodes, edges, choice1, choice2, game=2)
```

```
@app.callback(
    [Output('player11', 'children'), Output('player12', 'children')],
    [Input('choice1Player11', 'value'), Input('choice1Player21', 'value')]
)
def update_payoffs1(choice1, choice2):
    return calculate_payoff(1, choice1, choice2)
```

```
@app.callback(
    [Output('player21', 'children'), Output('player22', 'children')],
    [Input('choice1Player12', 'value'), Input('choice1Player22', 'value')]
)
def update_payoffs2(choice1, choice2):
    return calculate_payoff(2, choice1, choice2)
```

```
def calculate_payoff(game, a1, a2):
    if game == 1:
        if a1 == a2:
            return (1.25, 1.25) if a1 == 1 else (1.25, 1.25)
        elif a1 in {1, 2} and a2 in {1, 2}:
            return (1.75, 1.75)
        elif a1 in {2, 3} and a2 in {2, 3}:
            return (2.25, 2.25)
        elif a1 in {3, 4, 6} and a2 in {3, 4, 6}:
            return (2, 2)
        elif a1 == 7 or a2 == 7:
            return (2.25, 2.25)
        elif a1 == 5 and a2 == 5:
            return (1.5, 1.5)
```

```
    else:
        return (1.75, 2.25)
elif game == 2:
    if a1 == a2 and a1 != 2:
        return (1.5, 1.5) if a1 == 1 else (1.5, 1.5)
    elif a1 == 2 and a2 == 2:
        return (1.75, 1.75)
    elif a1 == 3 or a2 == 2:
        return (2.75, 2.75)
    elif a1 in {1, 2} and a2 in {1, 2}:
        return (2.25, 2.75) if a1 == 1 else (1.5, 1.5)
    elif a1 in {3, 4, 5, 6} and a2 in {3, 4, 5, 6}:
        return (2.5, 2.5)
    elif a1 == 7 or a2 == 7:
        return (2.5, 2.5)
    elif a1 in {8, 9} or a2 in {8, 9}:
        return (2.75, 2.75)
    else:
        return (1.75, 2.25)

if __name__ == '__main__':
    app.run_server(debug=True)
```

ДОДАТОК Б. ІЛЛЮСТРАТИВНИЙ МАТЕРІАЛ

Теоретико-ігровий підхід в задачі вибору локації

Студент IV курсу, групи КА-03

Ліфтеров Валентин Олександрович

Керівник:

доцент, к.ф.-м.н. Барановська Леся Валеріївна

- **Предмет дослідження:** застосування теорії ігор у сучасних практичних задачах
- **Об'єкт дослідження:** теоретико-ігровий підхід у задачі вибору локації
- **Мета дослідження:** створення програмного продукту для вирішення задач з вибору локації

Історія виникнення задачі з вибору локації



Рис. 1 П'єр де Ферма

Нехай дано три точки A , B і C на площині. Знайти таку точку P , що сума відстаней від цієї точки P до кожної з трьох точок A , B і C є мінімальною.

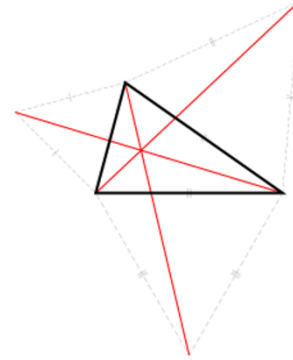


Рис. 2 Знаходження точки Ферма

Сучасна форма постановки задач з вибору локації

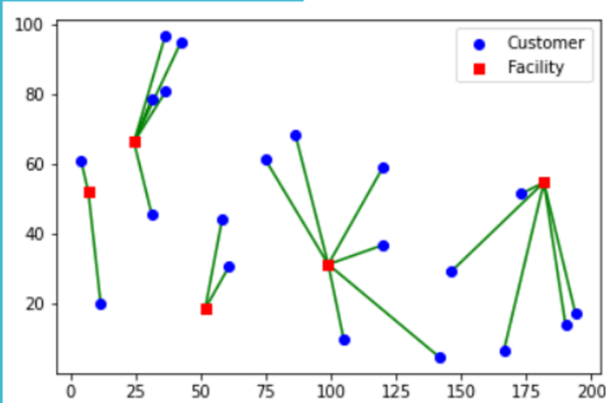


Рис.3 Приклад розв'язку задачі з вибору локації

1. Проблема визначення точки передачі (англ. The Transfer Point Location Problem), **TPLP**
2. Множинне розташування пунктів передачі (англ. The Multiple Location of Transfer Points), **MLTP**
3. Проблема розміщення об'єктів і пунктів передачі (англ. The Facility and Transfer Points Location Problem), **FTPLP**

Сфери застосування у сучасному світі

1. Logistics
2. Energy sector
3. Urban planning
4. Industrial production



Рис. 4 Логістика



Рис. 5 Енергетика

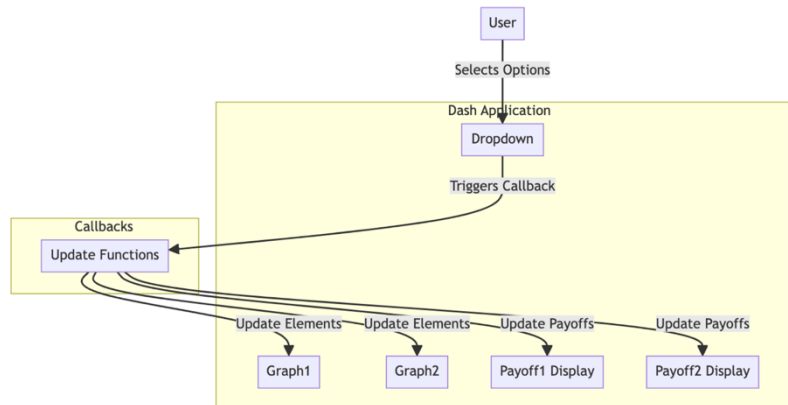
Програмний продукт. Умови задачі

1. Мережевий граф: Граф, що складається з вузлів і ребер, представляє область, де можуть бути розташовані поштові відділення. Кожне ребро має пов'язану вагу, яка представляє вартість маршрутизації.
2. Гравці: У грі беруть участь два гравці, кожен з яких обирає вузол у графі як місце для свого поштового відділення.
3. Вибір: Гравці незалежно обирають вузли, де вони хочуть розмістити свої поштові відділення. Вибори можуть бути однаковими або різними для кожного гравця.

Цілі гри яка реалізована в рамках програмного продукту

1. Мінімізувати витрати на маршрутизацію: Гравці прагнуть обрати місце для поштових відділень, які мінімізують суму к найбільших витрат на маршрутизацію у мережі.
2. Оптимізувати виграші: Кожен гравець прагне максимізувати свій виграш, стратегічно обираючи місце для поштового відділення, або централізуючи, або диференціюючи свої вибори на основі структури мережі.

Огляд програмного продукту



Приклади роботи програмного продукту

Гра у вибір локації поштового відділення

Гра 1	Гра 2
Вибір першого гравця	Вибір другого гравця
3	2
Виграші гравців	
Перший гравець: 2.25	Другий гравець: 2.25

Гра у вибір локації поштового відділення

Гра 1	Гра 2
Вибір першого гравця	Вибір другого гравця
2	3
Виграші гравців	
Перший гравець: 2.25	Другий гравець: 2.25

Приклади роботи програмного продукту

Гра у вибір локації поштового відділення

Гра 1	Гра 2
Вибір першого гравця	Вибір другого гравця
2	2
Виграші гравців	
Перший гравець: 1.75	Другий гравець: 1.75

Гра у вибір локації поштового відділення

Гра 1	Гра 2
Вибір першого гравця	Вибір другого гравця
3	2
Виграші гравців	
Перший гравець: 2.75	Другий гравець: 2.75

Використані ресурси:

1. Рис. 1. П'єр де Ферма. <https://ru.wikipedia.org/wiki/%D0%A4%D0%B5%D1%80%D0%BC%D0%B0,%D0%9F%D1%8C%D0%B5%D1%80>
2. Рис. 2. Знаходження точки Ферма. https://uk.wikipedia.org/wiki/%D0%A2%D0%BE%D1%87%D0%BA%D0%B0_%D0%A4%D0%B5%D1%80%D0%BC%D0%B0
3. Рис. 3. Приклад розв'язку задачі з вибору локації. <https://github.com/Jeevesh28/Facility-Location-Problem>
4. Рис. 4 Логістика. <https://www.fool.com/terms/l/logistics-in-business/>
5. Рис. 5 Енергетика. <https://arayara.org/renewable-energies-are-a-major-economic-breakthrough-for-the-global-energy-sector/>

Висновки

1. Було проведено дослідження контексту виникнення задач з вибору локацій як таких, персоналії людей які зробили внесок їх розвиток та практичні приклади їх постановок та вирішення в виді FLP, TPLP, MTPLP, FTPLP.
2. Описані основні принципи теорії ігор, в основі яких лежать парадигми кооперативної і некооперативної взаємодії гравців (учасників гри).
3. Створено програмний продукт, який дозволяє ознайомитись із теоретико-ігровим підходом до задачі вибору локації шлях розв'язання задачі про розміщення двох поштових відділень.



Дякую за увагу

