

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ

Факультет автоматизації інформаційних
технологій

Кафедра інформаційних технологій

(назва випускової кафедри)

ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ БАКАЛАВР

на тему:

Застосування теорії ігор для моделювання процесу прийняття
рішень у конфліктних ситуаціях.
Ч.1. Розробка концепції

Плісецький Ян Миколайович

Київ 2025 р.

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ**

автоматизації і інформаційних технологій

(факультет)

інформаційних технологій

(кафедра)

ЗАТВЕРДЖУЮ

Завідувачка кафедри ІТ

д.т.н., доцент Гончаренко Т.А.

„___” _____ 2025 року

**КВАЛІФІКАЦІЙНА РОБОТА
ЗДОБУВАЧА СТУПЕНЯ ВИЩОЇ ОСВІТИ БАКАЛАВР**

на тему: «Застосування теорії ігор для моделювання процесу
прийняття рішень у конфліктних ситуаціях.

Ч.1. Розробка концепції»

Я як здобувач вищої освіти КНУБА розумію і підтримую політику закладу з академічної доброчесності. Я не надавав(-ла) і не одержував(-ла) недозволену допомогу під час підготовки цієї роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

Здобувач

Плісецький Ян Миколайович

122 «Комп'ютерні науки»

(спеціальність)

Інформаційні управляючі системи і технології

(освітня програма)

Групи КН-21

Керівник Рябчун Ю.В.

(прізвище та ініціали)

Доктор філософії

(вчене звання, науковий ступінь)

Рецензент к.т.н., доц. Доля О.В.

(Прізвище та ініціали)

Ідентичність підтверджую

Київ, 2025 р.

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БУДІВНИЦТВА І
АРХІТЕКТУРИ**

Факультет:	Автоматизації інформаційних технологій
Випускова кафедра:	Інформаційних технологій
Освітній ступінь:	Бакалавр
Спеціальність:	Комп'ютерні науки
Освітня програма:	Інформаційні управляючі системи та технології

ЗАТВЕРДЖУЮ

Завідувач кафедри ІТ

Тетяна ГОНЧАРЕНКО

„___” _____ 2025 року

З А В Д А Н Н Я

ДО ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ ВИПУСКНОЇ РОБОТИ НА
ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ БАКАЛАВР

	Плісецькому Яну Миколайовичу
1. Тема роботи Застосування теорії ігор для моделювання процесу прийняття рішень у конфліктних ситуаціях. Ч.1. Розробка концепції	
затверджена наказом ректора КНУБА №235/23/25 від «14» лютого 2025 року	
2. Керівник роботи	Рябчун Юлія Володимирівна, PhD

3. Строк подання Здобувачем роботи до захисту _____

4. Зміст пояснювальної записки за розділами:

P.1 **АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ**

P.2 **ПРОЕКТ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

P.3 **КОНЦЕПЦІЯ РЕАЛІЗАЦІЇ ПРОГРАМНОГО ПРОДУКТУ**

P.4 **ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШОГО РОЗВИТКУ**

5. Графічний матеріал за розділами:

Р.1 1 рисунок

Р.2 1 рисунок; 4 таблиці

Р.3 3 рисунка; 1 таблиця

Р.4 1 рисунок

6. Календарний план виконання роботи:

Види робіт та їх зміст	Дата виконання
Розділ 1	17/02/2025
Розділ 2	04/04/2025
Розділ 3	10/05/2025
Розділ 4	27/05/2025
Остаточне оформлення роботи	28/05/2025
Направлення роботи для перевірки на плагіат	29/05/2025
Попередній захист роботи на випусковій кафедрі	30/05/2025
Направлення роботи на рецензування	30/05/2025

7. Консультанти розділів атестаційної випускної роботи

Розділ	Прізвище, ініціали та посада консультанта	Перевірив	
		дата	підпис
Розділ 1	Рябчун Ю.В., доцент каф.ІТ	17/02/2025	
Розділ 2	Рябчун Ю.В., доцент каф.ІТ	04/04/2025	
Розділ 3	Рябчун Ю.В., доцент каф.ІТ	10/05/2025	
Розділ 4	Рябчун Ю.В., доцент каф.ІТ	27/05/2025	

8. Дата видачі завдання січень 2025 р.

Зав. кафедри			Гончаренко Т.А.
	(підпис)		(прізвище та ініціали)
Керівники			Рябчун Ю.В.
	(підпис)		(прізвище та ініціали)
Здобувач			Плісецький Я.М.
	(підпис)		(прізвище та ініціали)

АНОТАЦІЯ

Плісецький Я.М. Застосування теорії ігор для моделювання процесу прийняття рішень у конфліктних ситуаціях. Ч.1. Розробка концепції.

Атестаційна випускна робота бакалавра за спеціальністю 122 «Комп'ютерні науки», освітня програма «Інформаційні управляючі системи та технології». – Київський національний університет будівництва та архітектури. – Київ, 2025.

Дана дипломна робота присвячена розробці програмного засобу для моделювання конфліктних ситуацій із застосування теорії ігор, зокрема в умовах конкурентної взаємодії двох компаній.

Дана дипломна робота присвячена розробці програмного засобу для моделювання конфліктних ситуацій, який дозволяє досліджувати поведінку двох гравців у конкурентному середовищі, аналізувати їхні стратегії, знаходити рівновагу Неша та оцінювати взаємодії. У роботі реалізовані такі функціональні можливості, як введення вхідних даних, побудови матриць виграшів, візуалізації ігрових моделей у вигляді блок-схем та дерев рішень. Експериментальна частина містить приклади моделювання конфліктних ситуацій, що дозволяють оцінити ефективність використання теорії ігор для прийняття рішень. На основі отриманих результатів сформульовано висновки та рекомендації щодо подальшого вдосконалення програмного продукту і розширення його функціональності для практичного застосування у сфері бізнесу та стратегічного планування.

Робота викладена на 81 аркуші, містить 5 таблиць, 6 рисунків, список використаної літератури із 25 найменувань.

Ключові слова: теорія ігор, прийняття рішень, конфліктна ситуація, гравці, нечітка логіка, моделювання, матрична гра, домінантні стратегії, нульова сума, кооперативні ігри, некооперативні ігри, вектор Шеплі, точка Вебера, алгоритм прийняття рішень, AnyLogic, Gambit, Python, TKinter.

SUMMARY

Plisetskyi Y.M. Application of Game Theory for Modeling Decision-Making Processes in Conflict Situations. Part 1. Concept development.

Bachelor thesis in specialty 122 "Computer Science", educational program "Information Control Systems and Technologies". – Kyiv National University of Construction and Architecture, Kyiv, 2025.

This thesis is devoted to the development of a software tool for modeling conflict situations, which allows you to study the behavior of two players in a competitive environment, analyze their strategies, find Nash equilibrium, and evaluate interactions. The work implements such functionalities as input data entry, construction of payoff matrices, visualization of game models in the form of flowcharts and decision trees. The experimental part contains examples of modeling conflict situations that allow us to evaluate the effectiveness of using game theory for decision-making. Based on the results obtained, conclusions and recommendations were formulated for further improvement of the software product and expansion of its functionality for practical application in the field of business and strategic planning.

Keywords: game theory, decision making, conflict situation, players, fuzzy logic, modeling, matrix game, dominant strategies, zero-sum, cooperative games, non-cooperative games, Shapley vector, Weber point, decision-making algorithm, AnyLogic, Gambit, Python, TKinter.

Зміст

ВСТУП	9
Розділ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ	11
1.1. Постановка та аналіз проблеми	12
1.2. Теоретичні основи теорії ігор	13
1.2.1 Визначення теорії ігор як наукової дисципліни	13
1.2.2 Історія розвитку теорії ігор: основні етапи та ключові особистості (Джон фон Нейман, Джон Неш)	13
1.3 Основні поняття теорії ігор	15
1.3.1 Гравці	15
1.3.2 Стратегії	15
1.3.3 Виграші	16
1.3.4 Матричні та коаліційні ігри	16
1.4 Типологія конфліктів у теорії ігор	18
1.4.1 Визначення конфліктних ситуацій.	19
1.4.2 Класифікація ігор залежно від типу конфлікту:	20
1.5 Основні математичні моделі в теорії ігор	21
1.5.1 Моделювання конфліктів	21
1.5.2 Моделі для прийняття рішень у багатокритеріальних умовах.	26
1.5.3 Застосування алгоритмів оптимізації для аналізу стратегій.	27
1.6 Концепція моделювання процесу прийняття рішень	29
1.7. Аналіз готових рішень	32
Розділ 2. ПРОЕКТ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	36
2.1. Моделювання процесів прийняття рішень у конфліктних ситуаціях. Аналіз конфліктних ситуацій	36
2.1.1. Характеристика конфліктів	36
2.1.2. Метод формалізації задач	37
2.2. Постановка моделі для конкретної конфліктної ситуації	38

2.2.1. Визначення гравців	39
2.2.2. Визначення цілей гравців	39
2.2.3. Обмеження гравців	42
2.2.4. Визначення стратегій для гравців	44
2.2.5. Формування матриці виграшів	45
2.2.6. Сутність матриці виграшів	45
2.3. Процедура формування	46
2.3.1. Визначення всіх можливих пар стратегій	46
2.3.2. Оцінка результатів взаємодії	46
2.3.3. Заповнення матриці	46
2.3.4. Формування матриці виграшів для конкретної ситуації – Конфлікт між двома компаніями конкурентами:	47
2.4. Застосування рівноваги Неша та інших підходів для пошуку оптимального рішення	48
2.5. Застосування методу домінування	50
2.6. Аналіз гри у змішаних стратегіях	51
2.7. Розробка програмної моделі для аналізу конфлікту двох компаній	52
2.7.1. Вхідні дані	52
2.7.2. Алгоритм вибору програмної реалізації	54
2.7.3. Інтерфейс користувача	54
2.7.4. Аналіз результатів моделювання.	55
2.7.5. Аналіз результатів моделювання.	56
Розділ 3. КОНЦЕПЦІЯ РЕАЛІЗАЦІЇ ПРОГРАМНОГО ПРОДУКТУ	58
3.1. Вибір інструментів розробки та обґрунтування	58
3.2. Архітектура програмного продукту	59
3.3. Опис функціональності програми	63
3.4. Модель даних та структура програми	64
3.5. Приклад сценарію використання програми	67

Розділ 4. ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШОГО РОЗВИТКУ	70
4.1. Загальні висновки щодо результатів дослідження	70
4.2. Оцінка функціональності реалізованого програмного продукту	71
4.3. Обмеження створеної моделі та програмного засобу	73
4.4. Перспективи подальшого розвитку програмного продукту	74
ВИСНОВОК	77
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	79

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І
ТЕРМІНІВ

- 1) **Теорія ігор** — математичний метод дослідження оптимальних стратегій поведінки у конфліктних або конкурентних ситуаціях, де результат залежить від дій усіх учасників (гравців).
- 2) **Прийняття рішень** — процес вибору найкращої дії серед кількох можливих варіантів, що ґрунтується на аналізі обставин, цілей і ризиків.
- 3) **Конфліктна ситуація** — ситуація, у якій інтереси двох або більше учасників (гравців) суперечать одне одному, і виграш одного може означати втрату іншого.
- 4) **Гравці** — учасники гри, кожен із яких має власні цілі, стратегії та можливості впливу на результат гри.
- 5) **Нечітка логіка** — метод обробки даних, що допускає багатозначність і невизначеність, дозволяє описувати складні системи з нечіткими (неповними або неточними) даними.
- 6) **Моделювання** — процес створення формалізованої моделі реальної ситуації для її дослідження, аналізу та прогнозування результатів у межах заданих умов.
- 7) **Матрична гра** — гра, у якій усі можливі стратегії двох гравців представлено у вигляді матриці, де кожна клітинка визначає виграш залежно від обраних стратегій обома сторонами.
- 8) **Домінантні стратегії** — стратегія, яка завжди дає гравцю кращий або не гірший результат незалежно від вибору опонента.
- 9) **Гра з нульовою сумою** — тип гри, в якій виграш одного гравця є точною втратою іншого. Сума виграшів усіх учасників завжди дорівнює нулю.
- 10) **Кооперативні ігри** — тип ігор, у яких гравці можуть об'єднуватися в коаліції, укладати угоди та діяти спільно для досягнення вигідних результатів.
- 11) **Некооперативні ігри** — ігри, у яких кожен гравець діє самостійно, не домовляючись із іншими учасниками. Домінують індивідуальні інтереси.

- 12) **Вектор Шеплі (вектор Шеплі-Вертуша)** — розподіл виграшу в кооперативній грі, який задовольняє принцип справедливості та враховує внесок кожного гравця в загальний результат.
- 13) **Точка Вебера** — оптимальне розташування точки (наприклад, об'єкта) у просторі, що мінімізує сумарну відстань до фіксованої множини інших точок, з урахуванням ваг.
- 14) **Алгоритм прийняття рішень** — чітка послідовність дій або кроків, яка веде до вибору найбільш оптимального рішення в умовах наявних даних і обмежень.
- 15) **AnyLogic** — універсальне середовище моделювання, що дозволяє будувати багаторівневі симуляції систем з використанням агентного, дискретного та системної динаміки підходів.
- 16) **Gambit** — програмне забезпечення для аналізу скінченних ігор у нормальній та розгорнутій формах. Дозволяє знаходити рівновагу Неша, домінантні стратегії та інше.
- 17) **Python** — популярна мова програмування, яка активно використовується в наукових, математичних і прикладних задачах, зокрема в реалізації алгоритмів теорії ігор.
- 18) **TKinter** — стандартна бібліотека Python для створення графічного інтерфейсу користувача (GUI), що використовується для побудови візуальних застосунків.

ВСТУП

Ми живемо у світі конфліктів. Кожного дня у людини виникають ситуації, коли потрібно зробити правильний вибір, аби досягти бажаного результату. Починаючи з побутового рівня – вибір кінофільму для перегляду разом з другом, закінчуючи важливими політичними процесами на світовому рівні. Ми можемо спостерігати застосування теорії ігор у художніх фільмах, таких як «Гра в імітацію», «Ігри розуму», «Матриця». Усі ці фільми об'єднує те, що головні герої використовують теорію ігор для успішної реалізації своїх планів (розшифрування коду, наукових досліджень чи боротьби з ворогами). Також гарним прикладом застосування теорії ігор є стосунки між батьками та дітьми. Кожна з сторін хоче максимізувати свій вииграш, враховуючи дії та реакції іншої сторони. У результаті ми бачимо, що незалежно від сфери застосування теорії ігор, правильно підібрана стратегія – це найважливіший аспект для здобуття гравцем вигоди для себе, або досягнення консенсусу у вирішенні конфлікту. Саме тому дослідження процесу прийняття рішень у конфліктних ситуаціях є дуже важливим для суспільства у наш час.

Дана дипломна робота присвячується дослідженню теорії ігор як інструменту моделювання та аналізу процесів прийняття рішень у конфліктних ситуаціях. Для усіх конфліктів, які можуть виникати у будь яких сферах життя, таких як економічна, політична, соціальна, мають бути розроблені ефективні підходи для аналізу ситуації, прогнозування тих чи інших подій та, як результат, вирішення конфлікту.

Метою роботи є розробка концепції, яка зможе виявляти ефективні підходи та мінімізувати конфлікти. У роботі буде детально розглянуто основи теорії ігор. Вони дозволять нам передбачати поведінку гравців, враховувати їхні інтереси та створювати найоптимальніші стратегії для досягнення компромісів.

Завдання дослідження:

1. Аналіз уже існуючих моделей вирішення конфліктів
2. Аналіз типів конфліктних ситуацій, які можна моделювати за допомогою теорії ігор

3. Проектування концептуальної моделі для аналізу та вирішення конфліктів.

4. Впровадження та тестування запропонованої моделі.

Об'єкт дослідження – процес прийняття рішень у конфліктних ситуаціях.

Предмет дослідження – концептуальна модель процесу прийняття рішень на основі алгоритмів теорії ігор.

Структура роботи – робота викладена на 81 аркуші, містить 5 таблиць, 6 рисунків, список використаної літератури із 25 найменувань.

Розділ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

В наш час конфліктні ситуації можуть відбуватись з людьми в будь який момент їхнього життя. Вони можуть чекати нас абсолютно у всіх сферах діяльності – соціальній, політичній, економічній, освітній, сімейній. Кожен з учасників того чи іншого конфлікту хоче виграти та отримати максимальну вигоду для себе, або в разі програшу відбутись малою кров'ю.

Не дивлячись на те, що існує безліч різних експертів у сферах конфліктів та різних стратегій для їх вирішення, конфлікти стають все складнішими. Тому не завжди вже існуючі методи можуть допомогти людям досягти поставленої цілі. Через еволюцію конфліктів потрібно покращувати самі стратегії вирішення проблем.

Аналіз предметної області – це один з найперших та важливих етапів розробки будь чого. Саме на даному етапі ми можемо більш детально зрозуміти специфіку взаємодії між учасниками конфліктів, а також зрозуміти їхні потреби. Це дозволяє обрати нам математичні моделі, стратегії та алгоритми, які будуть найкраще підходити для тих чи інших конфліктів, і допоможуть найбільш ефективно вирішити суперечності. Аналіз предметної області визначає напрямок подальшого дослідження та розробки моделі. Він допомагає визначити, які аспекти конфлікту (типи ігор, виграші, рівноваги) є найбільш критичними для прогнозування поведінки гравців та пошуку найбільш оптимальних рішень. Також за допомогою аналізу предметної області ми можемо виявити фактори, які будуть впливати на динаміку конфлікту. Це можна буде врахувати у моделюванні. Детальний аналіз предметної області допомагає знайти потенційні проблеми, які пов'язані з розробкою математичних моделей, формалізацією конфліктних ситуацій та вибором стратегій. Це дозволить мінімізувати ризики, забезпечити відповідність моделей і алгоритмів реальними умовам та потребам.

Постановка задачі допомагає сформулювати конкретні проблеми, які потребують вирішення, та цілі, які потрібно буде досягти. Це закладає фундамент

для подальшої розробки математичних моделей та алгоритмів, що будуть застосовані для вирішення конфлікту.

У цьому розділі буде проведено аналіз предметної області, що стосується розробки математичної моделі та різних алгоритмів для вирішення конфліктних ситуацій, аналіз вже готових прикладів моделей та алгоритмів, визначення основних плюсів та мінусів існуючих алгоритмів, а також конфліктна ситуація, яку необхідно вирішити за допомогою розробки алгоритмів та моделей вирішення конфліктних ситуацій.

1.1. Постановка та аналіз проблеми

Конфліктні ситуації стали невід'ємною частиною нашого життя. Вони присутні майже у всіх сферах діяльності – починаючи з соціальної сфери, закінчуючи політичною. Часто прийняття рішень в тих чи інших ситуаціях ускладнюється через такі аспекти, як [1]:

- Багато різних варіантів для дій;
- Інтереси учасників, які кардинально суперечать одне одному;
- Невизначеність щодо дій іншої сторони;
- Наявність певних ризиків які пов'язані з неправильними рішенням.

Класичні методи аналізу конфліктів, такі як інтуїтивні [2] або емпіричні [2] підходи, не завжди можуть допомогти нам формалізувати ситуацію та забезпечити об'єктивність вибору. Тому в таких випадках буде доречно та зручно використати теорію ігор як математичний інструмент для моделювання та аналізу конфліктних ситуацій [3].

Хоча і при застосуванні теорії ігор ми стикаємося з деякими проблемами, а саме [4]:

- Проблема з вибором підходящої моделі – будь-яка конфліктна ситуація матиме свої особливості, для яких потрібно буде застосувати різні типи ігор;
- Складність формалізації реальних конфліктів – потрібно конкретно визначити гравців, їхні стратегії та виграші;

- Урахування невизначеності та ризиків – учасники можуть не до кінця знати повну інформацію про «гру» або діяти ірраціонально;
- Інтерпретація результатів – потрібно чітко зрозуміти практичну значущість отриманих рішень та відповідне застосування їх в подальшій грі.

1.2. Теоретичні основи теорії ігор

В процесі вирішення складних конфліктних ситуацій, в яких взаємодіють багато сторін, виникає необхідність систематизувати процеси аналізу та розробки стратегій вирішення конфліктів. Один з ключових методів вирішення задач такого типу – це теорія ігор [2].

Теорія ігор – це теорія математичних моделей, які застосовуються для прийняття оптимальних рішень в конфліктних ситуаціях [3]. Вона використовується у багатьох сферах життя – соціальна, економічна, політична.

В даному пункті дипломної роботи будуть розглянуті теоретичні основи теорії ігор.

1.2.1 Визначення теорії ігор як наукової дисципліни

Теорія ігор – це розділ прикладної математики, а саме дослідження операцій. Зазвичай вона використовується в соціальних науках, а також в політичних, біологічних, тощо.

1.2.2 Історія розвитку теорії ігор: основні етапи та ключові особистості (Джон фон Нейман, Джон Неш)

Своєрідний приклад теорії ігор розглядався ще у XVIII столітті А. Курно та Ж. Бертран. Вони застосовували оптимальні рішення та стратегії моделювання для виробництва в умовах олігополії [5]. Іншим попередником вважається Е. Борель [6]. Також деякі фундаментальні ідеї теорії ігор були закладені А. Вальдом [7].

Головний недолік вищеназваних теорій полягав у тому, що економічні моделі розглядались «за ідеальних умов», тобто учасника турбувало лише збільшення власного прибутку, без урахування дій інших учасників. При цьому ігнорувався найголовніший фактор – конкуренція серед усіх учасників.

За вирішення цього питання взяли два математика-економіста – Оскар Моргенштерн та Джон фон Нейман. Дослідивши різні економічні моделі, дослідники зрозуміли, що діяльність учасника в економічній моделі схожа на гру проти інших гравців. Через це дослідники вирішили інтерпретувати економічну модель як гру, а всіх учасників зробити гравцями [2].

Саме у 1928 році на світ з'явилась стаття Джона фон Неймана «Про теорії суспільних ігор» [8]. Це зробило його офіційним основоположником теорії ігор. Вперше у цій статті було використано поняття «теорія ігор».

Через 12 років теорія ігор вперше була систематично викладена, а у 1944 році Джон фон Нейман та Оскар Моргенштерн випустили свою наукову працю «Теорія ігор і економічна поведінка» [9]. В цій роботі сформульовані основні поняття, такі як «гра», «виграш», «програш». Під час гри учасники можуть використовувати певні ресурси, взаємодіяти одне з одним. Основна мета – виграти, приймаючи рішення на основі вчинків інших гравців. Також було математично описано спосіб пошуку оптимальних стратегій для перемоги у грі.

Ще пізніше, теорію ігор почав розвивати американський математик Джон Неш. У 1949 році він написав дисертацію про теорію ігор [10]. Через сорок п'ять років він отримає за цю роботу Нобелівську премію з економіки.

Його внесок в теорію ігор полягав у фундаментальному аналізі рівноваги в теорії некооперативних іграх. Джон Неш значно розширив теорію ігор. Він допустив ситуацію, коли гравці не тільки можуть конкурувати між собою, а й кооперуватись для досягнення певної мети. На відміну від Неймана і Моргенштерна, які розглядали лише ігри з нульовою сумою, Джон Неш у своїх роботах представив аналіз «ігор з ненульовою сумою». В таких випадках учасники можуть або всі виграти, або всі програти. Саме так з'явилось поняття

«Рівновага Неша» - це сукупність стратегій, які допомагають учаснику досягти оптимального плану дій, спираючись на дії інших суперників.

1.3 Основні поняття теорії ігор

Після ознайомлення з історією виникнення теорії ігор, розглянемо основні поняття теорії ігор

1.3.1 Гравці

Гравець – це суб'єкт, який приймає рішення в межах гри [11]. Основна мета гравців – максимізація власного виграшу або досягнення поставленої мети. Гравцем не обов'язково є людина. Це може бути група осіб, компанія, держава, або будь яка інша система, яка робить стратегічний вибір. Кожен з гравців може мати набір стратегій, які буде застосовувати в залежності від того, які дії будуть виконувати противники. В більшості випадків гравці діють раціонально. Хоча це не завжди буває так.

Гравці мають можливість взаємодіяти між собою в залежності від гри – кооперативна (можуть співпрацювати) та некооперативна (конкурують між собою). Гравці можуть мати або повну інформацію про гру, або неповну (деякі аспекти залишаються невідомими).

1.3.2 Стратегії

Стратегія – це набір алгоритмів та правил, згідно з якими гравець діє у грі, враховуючи кроки опонентів [11]. Тобто стратегія буде визначати як гравець буде діяти у ігрових ситуаціях.

Існує декілька видів стратегій [11]:

1. Чиста стратегія – гравець завжди діє за одним і тим же принципом.
2. Змішана стратегія- гравець випадковим чином вибирає план дій відповідно до розподілу ймовірностей

3. Домінуюча стратегія-забезпечує гравцю найкращий результат, не залежно від дій інших гравців
4. Стратегія рівноваги Неша – жодному з гравців немає сенсу змінювати свою стратегію, якщо інші гравці не змінили свою
5. Оптимальна стратегія – мінімізація збитків кожного з гравців
6. Адаптивна стратегія – дії гравця змінюються під впливом дій інших гравців або досвіду з минулих ходів
7. Парето-ефективна стратегія – це стратегія, при якій дії іншого гравця не можуть покращити результат, без погіршення іншого гравця

1.3.3 Виграші

Виграш – це значення, яке характеризує результат кожного гравця залежно від обраної стратегії. Він може вимірюватись в балах, грошових одиницях, рейтингу, тощо [11].

Можна виділити декілька типів виграшів [11]:

1. Абсолютні виграші – конкретні числові значення, що отримують гравці.
2. Відносний виграш – виграш залежить від дій інших гравців.
3. Матричні виграші – представлені у вигляді матриці, де кожна клітинка – результат для гравців залежно від їхнього вибору.
4. У кооперативних іграх виграш може бути розподілено між гравцями відповідно до справедливого розподілу.
5. У некооперативних іграх кожен з гравців намагається максимізувати свій виграш.

Інколи виграш визначається на основі функції корисності, а не тільки кількісно. Ця функція відображає суб'єктивні переваги гравця (наприклад, в політиці це залежить від підтримки виборців)

Також гравці можуть використовувати певні стратегії для максимізації свого виграшу, такі як мінімаксна стратегія, домінантні стратегії та змішані стратегії.

1.3.4 Матричні та коаліційні ігри

Матричні ігри – це клас ігор, де виграші гравців представлені у вигляді матриць. Зазвичай цей тип гри застосовується для аналізу гри двох гравців [11]. Вони приймають рішення одночасно. Кожен з гравців має скінченну кількість стратегій. Наприклад, у гравця X є n кількість стратегій, у гравця Y є m кількість стратегій. Виграші кожного гравця представлені у вигляді матриці виграшів.

Існує декілька типів матричних ігор [11]:

1. Ігри з нульовою сумою – виграш одного з гравців дорівнює програшу іншого.
2. Ігри з ненульовою сумою – виграші гравців не обов'язково компенсують одне одного.

Є декілька варіантів розв'язання матричних ігор [11]:

1. Правило домінування – коли одна стратегія є вигідна, а друга ні, то другу стратегію можна вилучити.
2. Теорема фон Неймана – в іграх з нульовою сумою гравці намагаються мінімізувати свій максимальний програш. Розв'язком гри є сідлова точка.
3. Якщо сідлова точка відсутня – застосовується змішана стратегія. Гравці обирають стратегії з певними ймовірностями. Виграші можна визначити через лінійне програмування (симплекс – метод або алгоритм Брауна).

Існують також коаліційні ігри. Це такі ігри, в яких гравці можуть об'єднуватись в коаліції для спільного досягнення виграшу. На відміну від некооперативних ігор, гравці можуть домовлятися між собою про розподіл виграшів, розробляти спільні стратегії.

Кожна коаліція має певний виграш. Він визначається такою функцією :

$v(S)$, де $v : 2^N \rightarrow \mathbb{R}$. Ця функція відображає який виграш може отримати та чи інша коаліція в залежності від дій гравців [11].

Коаліційні ігри бувають кількох типів:

1. Суперадивні ігри (ігри з підсумовуванням виграшу). Для будь який коаліцій виконується таке рівняння: $v(S_1) + v(S_2) \leq v(S_1 \cup S_2)$. Це означає, що коаліціям вигідно об'єднуватись тільки тоді, коли їх загальний виграш буде більшим, ніж сума виграшів окремих груп.

2. Ігри з пороговими виграшами – у таких іграх коаліція може досягти виграшу тільки при умові достатньої кількості учасників

3. Нульова та ненульова сума. Нульова сума – коли виграш однієї коаліції дорівнює програшу іншої. Ненульова сума – всі учасники можуть отримати вигоду від коаліції.

Також є методи для розподілу виграшу в коаліційних іграх:

1. Принцип рівного розподілу – кожен з учасників коаліції отримує однаковий виграш

2. Шеплі-значення – кожен з учасників отримує виграш пропорційний до його внеску в гру.

3. Основи ядра гри. Ядро гри – це множина розподілів виграшу, за яких жодна коаліція немає стимулу виділятися. Якщо ядро порожнє – стабільний розподіл неможливий

1.4 Типологія конфліктів у теорії ігор

Конфлікти є невід'ємною частиною майже усіх сфер життя – економічної, політичної, соціальної, тощо. Моделювання цих конфліктів за допомогою теорії ігор допомагає більш детально зрозуміти закономірності ухвалення рішень. У теорії ігор, конфлікт – взаємодія гравців, які мають певні цілі та приймають рішення на основі власних стратегій для досягнення цих цілей.

Класифікація конфліктів в рамках теорії ігор дозволяє структурувати їх за різними критеріям, наприклад: кількість гравців, можливість кооперування, характер взаємодії, специфіка розподілення виграшів. Таким чином ми можемо краще підібрати стратегії для гравців.

В даному розділі буде розглянуто типи конфліктів, підходи до їхнього аналізу та розв'язання поставлених задач.

1.4.1 Визначення конфліктних ситуацій.

Конфліктна ситуація – це така ситуація, коли двоє або більше учасників мають різні інтереси і повинні ухвалювати певні рішення, які по впливатимуть на їхній стан та стан противників [12]. В теорії ігор моделювання конфлікту відбувається через демонстрацію різних дій гравців, які хочуть максимізувати свій виграш.

Можна виділити кілька основних характеристик конфлікту:

- Наявність гравців
- Розходження інтересів
- Обмежені ресурси
- Невизначеність

Якщо говорити про теорію ігор, то в ній ситуації класифікуються за такими критеріями:

1. Кількість учасників:

- Парні конфлікти – учасники це дві сторони
- Багатосторонні конфлікти – три і більше

2. За можливістю кооперації:

- Кооперативні – гравці можуть створювати коаліції
- Некооперативні – такої можливості немає

3. За розподілом виграшів:

- Ігри з нульовою сумою – виграш одного гравця дорівнює програшу

другого

- Ігри з ненульовою сумою – виграш одного гравця необов'язково

дорівнює програшу другого

4. За динамікою прийняття рішень:

- Статичні конфлікти – гравці одночасно приймають рішення, не знаючи рішення інших учасників
- Динамічні конфлікти – гравці поступово приймають рішення, враховуючи свої попередні ходи та ходи противників.

1.4.2 Класифікація ігор залежно від типу конфлікту:

В даному підпункті ми більш детально розглянемо типи конфліктів:

Ігри з нульовою сумою. Ігри з нульовою сумою – один із типів стратегічних ігор в теорії ігор [11]. Їхня суть полягає в тому, що вигреш одного гравця дорівнює втратам іншого. Загальна сума вигравів завжди дорівнює нулю.

Ігри з нульовою сумою мають конкурентний характер. Кожен з учасників діє виключно з розрахунком на власну вигоду. Учасники не можуть кооперуватись між собою. Загальна сума вигравів та програвів є сталою. Неможливо створити нові виграти без втрат іншої сторони. Стратегії одного гравця безпосередньо визначають результат іншого гравця. У таких іграх зазвичай використовується рівновага Неша або мінімаксна стратегія. Їх використовують для того, щоб мінімізувати максимальні втрати. Прикладом таких ігор є шахи, спортивні змагання, азартні ігри.

Для розв'язання таких ігор використовують різні методи. Наприклад, мінімаксний принцип фон Неймана – гравець обирає таку стратегію, яка мінімізує його втрати. Також можна використати рівновагу Неша. Ще можна застосувати методи лінійного програмування або графічний метод.

Кооперативні та некооперативні ігри. Конфлікти можна також класифікувати за можливістю гравців домовлятися між собою. Ігри, в яких гравці можуть об'єднуватися в коаліції називаються кооперативні. Учасники конфлікту можуть укласти між собою угоди, які будуть зобов'язувати їх до виконання певних дій та обов'язків. Відповідно до цього у центрі уваги стоїть вигреш всієї коаліції, а не лише одного гравця. Вигреш можуть розподіляти, наприклад, за

допомогою Шеплі-значень – гравець отримує вигреш пропорційний до його вкладу в гру.

Некооперативні ігри – це ігри в який гравці діють самостійно та не можуть укладати між собою угоди [11]. Головна мета кожного гравця – максимізувати свій вигреш або мінімізувати прогреш, враховуючи дії опонентів. Таким іграм властивий більш конкурентний характер, так як кожен працює тільки на власне благо. Основний концепт таких ігор – рівновага Неша. Жоден гравець не може покращити свій вигреш, якщо інші гравці не змінюватимуть свої стратегії. Стратегії можуть бути чистими або змішаними.

Симетричні та асиметричні ігри. Не менш важливою є класифікація ігор за принципом симетрії. За допомогою цього ми краще аналізуємо стратегічну взаємодію між гравцями.

Симетрична гра – це гра, в якій вигреші гравців залежить від обраних стратегій, а не від того хто їх обрав [11]. В симетричних іграх усі гравці знаходяться в однакових умовах – вони всі мають однаковий набір стратегій. Ніхто не має переваги над іншими гравцями. У багатьох випадках оптимальні стратегії для гравців є однакові. Чудовим прикладом симетричних ігор є «Дилема в'язня». Кожен з в'язнів мають однакові стратегії – або мовчати, або зізнатись. Результат гри залежить тільки від послідовності дій гравців і поєднання їх вибору.

Якщо говорити про несиметричні ігри, то це ігри, де у гравців різні умови і вигреш залежить не тільки від обраних стратегій, а і від тих, хто ці стратегії використовує. Тобто в несиметричній грі навіть якщо гравці обирають одні і ті ж стратегії, функція вигрешу буде різна для кожного гравця. В несиметричних іграх гравці можуть мати різний набір стратегій. В іграх даного типу є слабші і сильніші сторони. Оптимальні стратегії для кожного з гравців можуть суттєво відрізнатись. Також в гравців може бути різний контекст – один може бути виробником, інший споживачем.

Якщо звести усі основні характеристики ігор в одну діаграму, матимемо рисунок 1.1.

1.5 Основні математичні моделі в теорії ігор

1.5.1 Моделювання конфліктів

У теорії ігор моделювання конфліктів дозволяє формалізувати складні конфліктні ситуації. Завдяки цим математичним моделям можна не лише зручно описувати взаємодію гравців, а й прогнозувати їхні можливі ходи та результати. Одним з основних інструментів аналізу матричні ігри. Вони дозволяють наочно показати зв'язки між стратегіями та виграшами. Також часто застосовують концепцію рівноваги Неша. Вона дозволяє визначити умови стабільності у конфліктних ситуаціях. Ці підходи є основними для розуміння процесу прийняття рішень у конфліктних умовах і знаходження оптимальних стратегій для всіх учасників гри.

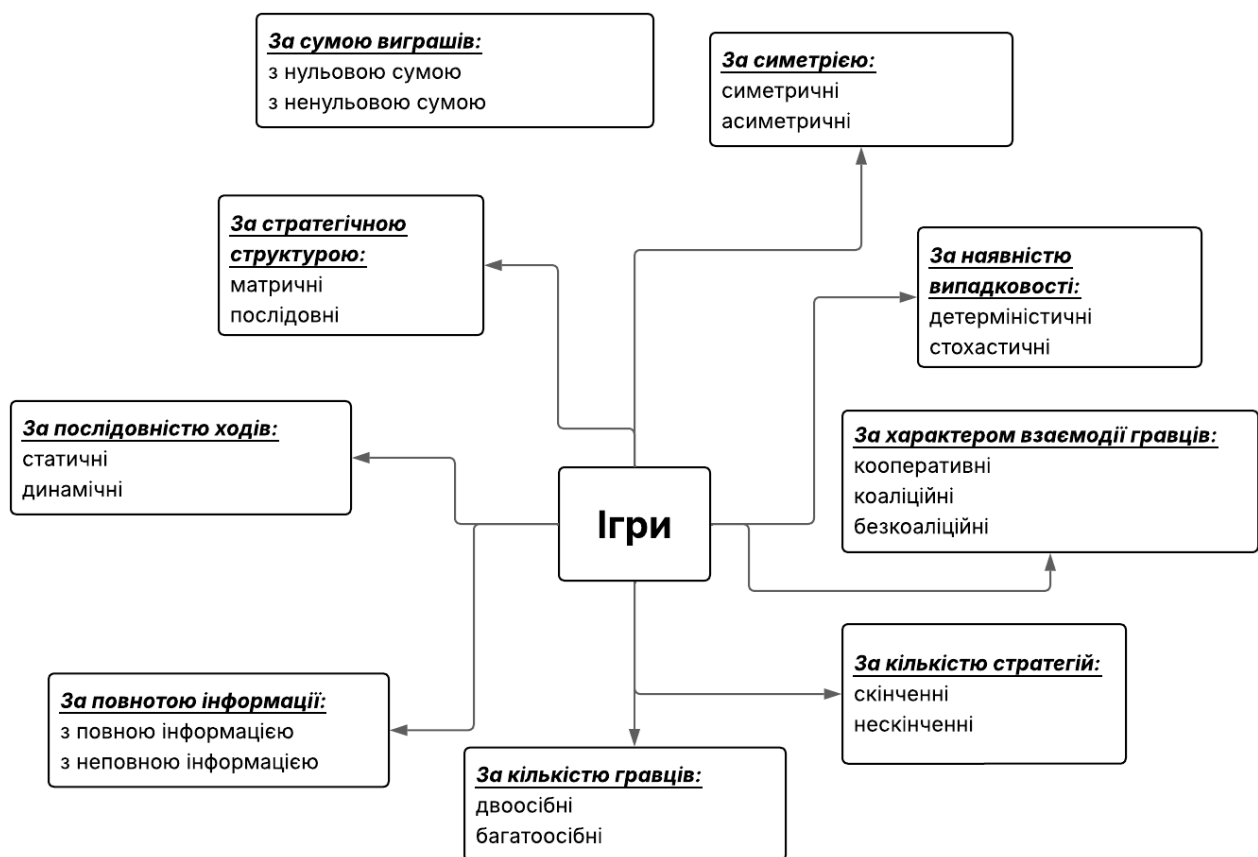


Рисунок 1.1 – Основні характеристики ігор в одній діаграмі

Матричні ігри та їх розв'язання. Матричні ігри – це клас ігор, які описуються за допомогою матриці [11]. В цій матриці зазначені усі можливі

стратегії гравців і відповідні виграші для кожної комбінації цих стратегій. Зазвичай такі ігри моделюють конфліктні ситуації між двома гравцями з кінцевою кількістю стратегій. Існує декілька основних типів матричних ігор. Перший тип це ігри з нульовою сумою. Суть цих ігор полягає в тому, що виграш одного гравця дорівнює програшу іншого гравця. Тобто сума виграшів обох гравців дорівнює нулю. Також існують ігри з ненульовою сумою. В цих іграх сума виграшів не обмежена нулем. Гравці можуть як конкурувати, так і співпрацювати одне з одним. Існує декілька методів розв'язання матричних ігор.

Одним із них є метод домінування. Суть даного методу полягає у поступовому виключенні стратегій, які є найменш оптимальними, ніж інші стратегії, не залежно від дій інших гравців. Також є метод мінімаксу. Він полягає в тому, щоб гравець обрав для себе стратегію яка у випадку програшу забезпечить йому найкращий результат. Якщо значення для обох гравців будуть збігатись, то буде отримана сідлова точка – рівновага гри. Також часто застосовують розв'язання через змішані стратегії. Якщо чисті стратегії не дають гравцям бажаного результату, то використовують змішані стратегії. Це ймовірнісний розподіл над множиною чистих стратегій. Оптимальна стратегія визначається за допомогою системи рівнянь, що описують ймовірності вибору стратегій для досягнення рівноваги Неша.

$$p * A_{11} + (1 - p) * A_{21} = p * A_{12} + (1 - p) * A_{22}$$

Аналіз рівноваги Неша. Рівновага Неша – це набір стратегій у грі, коли кожен гравець робить так, як буде зручно йому. При цьому, він опирається на дії інших гравців. Слід зауважити, що ніхто не може покращити свій стан сам по собі, не враховуючи дії решти.

Математична формалізація звучить так: Ситуація $x^8 = (x_1^*, x_2^*, \dots, x_n^*)$ називається ситуацією рівноваги по Нешу, якщо для усіх $x_i \in X_i \quad i \in N$ справедлива нерівність:

$$K_i(x^*, x^*) > K(x, x^*)$$

Існує два типи рівноваги Неша [11]:

- Чиста стратегія – гравець однозначно визначає свою стратегію
- Змішана стратегія – гравець обирає стратегію з певною ймовірність

Рівновага Неша використовується у багатьох сферах життя, таких як:

- Військова справа – планування дій мій сторонами конфлікту
- Економічні та бізнес сфери – аналіз поведінок постачальників та споживачів
- Політика – планування та моделювання переговорів

Якщо говорити про переваги рівноваги Неша, можна виділити такі аспекти:

- Об'єктивність – можливість передбачати раціональні дії гравців
- Універсальність – рівновага Неша може застосовується у великій кількості ігор
- Математична чіткість – можна опрацьовувати складні конфлікти та враховувати усі фактори

- Можливість прогнозування наслідків вибору тієї чи іншої стратегії

Розглянемо недоліки рівноваги Неша:

- Складність обчислення – можливість обробки великої кількості інформації про гравців та гру створює досить об'ємні обчислення, які є не дуже зручними
- Ідеалізація припущень – рівновага Неша передбачає те, що усі гравці матимуть повну інформацію та будуть діяти раціонально. Це не завжди може бути так.
- Обмеженість факторів – у реальних конфліктах рівновага Неша не бере до уваги багато аспектів, які можуть сильно по впливати на подальший розвиток гри
- Відсутність стабільності – у динамічних система рівновага Неша може бути нестабільною. Також додаткові зовнішні фактори теж можуть порушувати стабільність
- Деякі ігри можуть мати декілька точок рівноваги Неша. Це може ускладнити вибір результату

Підводячи підсумки можна сказати що рівновага Неша це дуже зручний інструмент для моделювання процесу прийняття рішень у конфліктних ситуацій. Ми можемо обробляти та аналізувати багато інформації про гру.

Вектор Шеплі. Вектор Шеплі – це принцип розподілу виграшів між гравцями, в залежності від їх середнього вкладу у виграш [11].

Математичне формулювання даного принципу звучить так: Вектор Шеплі кооперативної гри називається такий розподіл виграшу, що кожен гравець отримує математичне сподівання свого вкладу у відповідній коаліції K_i , при рівноймовірному виникненні впорядкувань:

$$\Phi(v) = \frac{1}{n!} \sum_{\tau \in T} x_{\tau}$$

де n – кількість гравців, T – множина впорядкувань множини гравців N , x_{τ} – розподіл виграшу, в якому гравець, що стоїть на місці i у впорядкуванні τ , отримує вклад в коаліцію K_i (точка Вебера).

Більш розповсюджена формула, в якій не потрібно шукати точки Вебера має вигляд:

$$\Phi(v) = \sum_{K \ni i} \frac{(k-1)!(n-k!)}{n} (v(K) - v(\frac{k}{i}))$$

де n – кількість гравців, k – кількість учасників коаліції K .

Коефіцієнт Шеплі враховує для кожного гравця, скільки виграшу він принесе з кожної можливої коаліції. Головна суть – розподілити виграш для кожного гравця відповідно до його важливості.

Даний принцип можна використовувати в багатьох сферах. Наприклад:

- Бізнес – розподіляти справедливо прибуток між партнерами, залежно від їхнього вкладу в компанію
- Спорт – можна розподіляти нагороди для спортсменів, в залежності від їхніх результатів
- Політика – розподіл голосів серед різних партій, залежно від їхнього розміру та вкладу в політичні аспекти

Розглянемо переваги принципу Шеплі:

- Об'єктивність – Шеплі-коефіцієнт базується на прямих розрахунках. Це забезпечить об'єктивність та допоможе уникнути дискримінації
- Універсальність – даний принцип можна застосовувати в будь-яких конфліктних ситуаціях

- Справедливість – даний принцип правильно розподіляє виграш кожного з гравців, пропорційно їхньої важливості або вкладу

Розглянемо недоліки даного принципу:

- Складність обчислення – у великих конфліктах з великою кількістю гравців дуже складно на основі вкладу та важливості кожного обчислити скільки має отримати гравець в результаті
- Часто в іграх гравці можуть використовувати різні стратегічні маніпуляції та хитрощі. Це не враховується при обчисленні вектора Шеплі та може негативно впливати на кінцевий результат
- Існує також проблема диспропорцію у невеликих коаліціях. Так як внески гравців можуть суттєво відрізнятися одне від одного і через це не завжди отриманий результат буде найоптимальнішим

Як висновок можна сказати, що принцип Шеплі – це зручний інструмент для розподілу виграшу між гравцями. Майже у всіх випадках виграш буде розподілений чесно. Проте в реальних конфліктах це може бути не завжди зручно або справедливо, бо неможливо врахувати деякі аспекти.

1.5.2 Моделі для прийняття рішень у багатокритеріальних умовах.

В реальних конфліктах зазвичай рішення базуються на багатьох факторах, які потрібно враховувати. Часто ці фактори бувають суперечливими. В таких випадках гравці застосовують багатокритеріальне прийняття рішень. Це дозволяє оцінити усі можливі варіанти розвитку подій з урахуванням усіх важливих аспектів.

Багатокритеріальне рішення – це вибір оптимальної стратегії, що допоможе отримати найкращий баланс між усіма цілями. В теорії ігор такі ситуації моделюються за допомогою спеціальних математичних підходів, що дозволяють гравцям врахувати вплив багатьох факторів.[13]

Одним з основних підходів до моделювання багатокритеріальних рішень це метод скаляризації критеріїв. Цей метод передбачає об'єднання всіх критеріїв в одну функцію. Це неабияк спрощує аналіз гри. Сама функція виглядає так

$$U(x) = w_1 * f_1(x) + w_2 * f_2(x) + \dots + w_n * f_n(x)$$

$f_i(x)$ – функції, що описують окремі критерії, а w_i – вагові коефіцієнти, які відображають важливість кожного з критеріїв.

Даний метод досить простий в обчисленнях. Але існує складність у визначення вагових коефіцієнтів. Нерідко це призводить до суб'єктивності.

Також існує метод аналізу ієрархій. Його розробив Т.Сааті. Структура даного методу складається з декількох пунктів [14]:

1. Визначення мети прийняття рішення
2. Побудова ієрархії критеріїв та альтернатив
3. Парне порівняння критеріїв за важливістю
4. Обчислення пріоритетів і визначення найкращої альтернативи

Це все обчислюється через матрицю попарних порівнянь. Це дозволяє найбільш оптимально оцінити важливість кожного критерію.

Ще існує концепція Парето-оптимальності. Вона використовується для опису рішень, коли за рахунок покращення одного критерію, погіршується інший. Як приклад можна навести гру, коли існує дві стратегії – одна більш прибуткова і більш ризикована, а інша навпаки менш прибуткова та менш ризикова. Кожна з цих стратегій може бути Парето – оптимальна, в залежності від пріоритетів кожного з гравців.

1.5.3 Застосування алгоритмів оптимізації для аналізу стратегій.

Під час вирішення складних конфліктних ситуацій, коли є багато учасників, великий об'єм усіх можливих стратегій та критеріїв, виникає необхідність застосовувати ефективні методи для знаходження оптимальних рішень. Це можна реалізувати за допомогою алгоритмів оптимізації. Вони дозволяють

систематизувати процес аналізу стратегій. У теорії ігор оптимізаційні алгоритми застосовують для:

- Пошуку рівноважних стратегій
- Мінімізація ризиків
- Максимізація виграшів
- Аналіз компромісних рішень

Одним із алгоритмів оптимізації у теорії ігор є лінійне програмування. Він дозволяє знаходити максимум або мінімум лінійної цільової функції при наявності обмежень. Часто цей підхід використовують для аналізу матриць або двохосібних ігор з нульовою сумою. Тобто, за допомогою лінійного програмування ми можемо визначити оптимальну стратегію для гравця, знайти рівноважний виграш в іграх з нульовою сумою та оптимізувати ресурси в економічних моделях.

Наступним алгоритмом є динамічне програмування. Це метод розв'язання складних задач, які можна розбити на підзадачі. Даний метод є досить ефективним для аналізу багатокрокових ігор, де всі рішення приймаються поступово. Тобто оптимальне рішення складається з оптимальних рішень для підзадач.

Також існують генетичні алгоритми. Вони імітують процес природнього відбору. Даний метод є ефективним в тих випадках, коли класичні методи мають обмеження. Існують такі етапи генетичних алгоритмів:

1. Ініціалізація
2. Оцінка
3. Селекція
4. Кросовер і мутації
5. Ітерція

Цей алгоритм використовується в пошуках глобального оптимуму в складних багатовимірних задачах, розробці стратегій для багатосторонніх переговорів, оптимізації ресурсів у динамічному середовищі.

Ще один алгоритм, який можна використати для оптимізації аналізу стратегій це алгоритм найкращої відповіді. Даний алгоритм використовується для того, щоб знайти рівновагу Неша. Суть алгоритму заключається в тому, що кожен

гравець обирає стратегію, яка є найкращою відповіддю на дії інших гравців. Цей алгоритм застосовується для аналізу економічної конкуренції, стратегічному планування, визначення оптимальних стратегій в соціальних мережах. Основні етапи алгоритму є:

1. Ініціалізація з випадкової стратегії
2. Послідовне оновлення стратегії кожного гравця для максимізації виграшу
3. Перевірка наявності рівноваги Неша
4. Повторення усіх пунктів до стабільності рішень

1.6 Концепція моделювання процесу прийняття рішень

Постановка задачі моделювання. Потрібно розробити математичну модель для вирішення конфліктних ситуацій, яка дозволяє досліджувати стратегії, що обираються учасниками конфлікту, а також визначити оптимальні стратегії.

Вхідні дані:

- Інформація про учасників конфлікту (кількість учасників, доступні стратегії, можливі типи учасників, мета)
- Інформація про ресурси та обмеження
- Дані для моделювання поведінки учасників
- Інформація про взаємодію між учасниками
- Інформація про можливі стратегії

Вихідні дані:

- Оптимальні стратегії
- Результати моделювання конфлікту
- Рекомендації по стратегіям
- Визначення рівноваги Неша
- Аналіз результатів

Функції системи:

- Введення даних
- Моделювання конфлікту

- Аналіз рівноваги
- Оптимізація стратегій
- Виведення результатів
- Аналіз ризиків

Навчання та адаптації на основі попередніх рішень або конфліктів

Дерево цілей. Дерево цілей – це графічний інструмент, який ієрархічно представляє основну проблему, усі підпункти проблеми, які показують шляхи вирішення даної проблеми [15]. Дерево цілей дозволяє:

- Візуалізувати головну проблему
- Розбити основну ціль роботи на простіші підпункти
- Побачити зв'язки між цілями
- Визначити пріоритети серед усіх напрямків роботи

Головна мета дипломної роботи – розробити ефективний підхід для моделювання процесу прийняття рішень у конфліктних ситуаціях за допомогою теорії ігор. Для того щоб досягнути поставленої задачі, потрібно розробити декілька рівнів підпунктів, кожен з яких буде спрямований на вирішення певних аспектів проблеми.

Першим кроком буде дослідження основних принципів теорії ігор. Це буде базою для моделювання конфліктних ситуацій. Саме на цьому етапі потрібно проаналізувати тип ігор (кооперативні та некооперативні). Також потрібно розглянути концепцію рівноваги Неша та її варіацій. Особливо детально потрібно опрацювати ігри з повною та неповною інформацією. В реальному житті конфлікти часто бувають з неповними даними про дії та наміри гравців.

Наступний крок - розробка математичної моделі конфліктної ситуації. Потрібно визначити основних гравців, цілі, усі можливі стратегії, та взаємодії. Особливо детально потрібно змоделювати виграші та втрати усіх учасників. Також треба оцінити точність та ефективність вибраних методів, щоб визначити чи будуть вони придатні для вирішення поставлених задач.

Після потрібно проаналізувати реальні приклади конфліктних ситуацій. Потрібно взяти декілька випадків з реального життя з різних сфер життя, сформувавши з них математичні моделі та оцінити отримані результати.

Після аналізу розробляються стратегії управління конфліктами. Їх можна буде застосувати в реальному житті. Потім потрібно підготувати рекомендації для використання моделі у різних сферах життя. Проаналізувавши ефективність рекомендацій можна буде оцінити реальну користь у вирішенні конфліктів.

Вимоги та особливості проектування системи. Даний розділ дипломної роботи є одним з ключових етапів у розробці будь-якого інформаційного проекту. Ці особливості та вимоги формують параметри та характеристики системи, які спрямовані на задоволення потреба користувачів та вирішення поставлених завдань.

Вимоги до системи:

1.6.1. Функціональні вимоги:

- Створення математичної моделі конфліктної ситуації
- Розрахунок рівноваги
- Моделювання сценаріїв з повною та неповною інформацією
- Автоматизація аналізу даних
- Генерація звітів
- Візуалізація моделі та процесу прийняття рішень
- Підтримка багатокритеріального аналізу

1.6.2. Безпека та конфіденційність

- Анонімність даних про гру
- Обмежений доступ до моделі та результатів
- Захист даних про гравців та гру

1.6.3. Інтерфейс

- Зрозумілий, зручний дизайн
- Зручність роботи з даними
- Візуалізація

- Комфортне налаштування інтерфейсу
- Сповіщення про помилки
- Навчання та допомога

1.6.4. Технології

- Штучний інтелект
- Математичні симуляції та моделювання
- Різні бази для зберігання даних
- Сервіси для візуалізації

1.6.5. Розширюваність

- Створення моделі з урахуванням подальшого покращення та змінення.

1.6.6. Продуктивність

- Оптимізація алгоритмів для розв'язання задач
- Мінімізація часу для обробки конфліктів

1.6.7. Інтеграція

- Сумісність з різними платформами, а також можливість взаємодії з

хмарними сервісами

1.6.8. Модульність

- Модель буде складатись з модулів, які можна буде або покращити або

замінити

1.6.9. Моніторинг та підтримка

● Потрібно реалізувати систему моніторингу і підтримки користувачів, щоб у разі чого можна було швидко виправити помічені користувачами недоліки

- *Визначення учасників та їхніх цілей у конфліктній ситуації.*
- *Вибір математичної моделі для аналізу.*

1.7. Аналіз готових рішень

Для того, щоб розробити зручну та практичну модель для вирішення конфліктних ситуацій, потрібно проаналізувати вже готові теорії. Розібравши

сильні і слабкі сторони кожної з теорій, можна виявити моменти, які можна буде застосувати для розробки нашої моделі.

Це допоможе більше зрозуміти що буде більш комфортним для гравців, що навпаки є не дуже вдалою реалізацією. Саме через це наша модель зможе бути конкурентоспроможною серед усіх існуючих моделей.

Для початку розглянемо існуючі теоретичні рішення. Це будуть моделі, які використовуються для вирішення конфліктних ситуацій.

Gambit – це програмне забезпечення, як використовується для аналізу і моделювання конфліктних ситуацій [16]. Він дозволяє виконувати багато обчислень, таких як згадана вище рівновага Неша, симулювати різні ігри та проводити їх аналіз.

Він підтримує багато різних видів ігор (з повною чи неповною інформацією, кооперативні ігри, змішані стратегії, тощо). Програма дозволяє користувачам моделювати різні ігри. Одна з основних функцій програми – рівновага Неша. Gambit може здійснювати розрахунки різної складності. Він має зрозумілий інтерфейс, який дозволяє гравцям корегувати ігри, а також візуалізувати результати та стратегії.

Плюси даного програмного забезпечення:

- Gambit має досить широкий функціонал. Він дозволяє обчислити будь які розрахунки, незалежно від складності
- Програма використовує зручні та сучасні алгоритми. Це дозволяє ефективно опрацьовувати дуже об'ємні ігри
- Gambit може опрацьовувати з багатьма видами стратегій
- Програма дає можливість візуалізувати обрані стратегії, отримані результати. Це допомагає гравцям краще зрозуміти процес гри
- Gambit є досить гнучким. Він може співпрацювати з багатьма різними програмними продуктами

Із мінусів можна виділити такі пункти:

- Потрібно мати потужну техніку для того щоб можна було запускати великі ігри з складними обчисленнями

- Не дивлячись на те, що інтерфейс інтуїтивно зрозумілий, для початківців може бути складно спочатку зрозуміти як працювати з цим програмним забезпеченням, так як воно містить в собі безліч функцій та алгоритмів
- Gambit має обмеження для великих ігор. Через об'ємні обчислення може зрости час обробки результатів. Через це можуть бути застосовані обмеження
- Інколи Gambit може не справлятися з великою кількістю стратегій та багатьма обчисленнями

Як підсумок можна сказати, що Gambit є досить зручним та сучасним засобом для аналізу та генерації конфліктних ситуацій. Проте можуть виникнути проблеми з роботою з об'ємними іграми, які міститимуть в собі велику кількість стратегій та математичних обчислень.

AnyLogic – це програмне забезпечення, яке моделює складні системи. Воно використовується для оптимізації багатьох процесів, а також аналізу стратегій та результатів у різних сферах життя [17].

Програма має багато систем моделювання, такі як Agent-Based Modeling. У цій системі кожен елемент системи має свої властивості, правила взаємодії та поведінки.

Discrete Event Simulation – описує процеси як ерії подій, які відбуваються в певний проміжок часу.

System Dynamics – моделює зміни в системах і дозволяє відстежити та проаналізувати динаміку зв'язків між системами.

Також AnyLogic дозволяє комбінувати різні моделі.

Програма має графічний інтерфейс. Це дозволяє користувачам будувати моделі без програмування. Середовище є досить зрозумілим. Також програма дозволяє моделювати різні стратегії, змінювати та оптимізувати параметри для найкращого застосування. AnyLogic може працювати у зв'язці з іншими системами для обробки даних. Це дозволяє більш детально пропрацювати певні аспекти та отримати більш точні результати.

Розглянемо плюси AnyLogic:

- Підтримує три основні методи моделювання, що робить її досить універсальною
- Графічний інтерфейс допомагає програмі бути більш доступною для користувачів, які не вміють програмувати
- AnyLogic є досить гнучкою. Її можна комбінувати з різними інструментами, що полегшить роботу
- Досить велика кількість вбудованих інструментів
- AnyLogic може справлятися навіть з дуже великими та складними моделями. Це неабияк розширює можливості користувачів

Мінуси програми:

- Так як AnyLogic є комерційним програмним забезпеченням з розширеним функціоналом – то і ціна за неї є досить високою
- Не дивлячись на те, що за допомогою графічного інтерфейсу користувачі можуть створювати різні моделі, при потребі розробки більш складніших моделей потрібно буде освоїти Java
- Через дуже широкий спектр функцій, програма потребує обладнання з високою потужністю
- Якщо порівнювати з іншими програмами, AnyLogic має недостатньо велику кількість стандартних бібліотек та шаблонів для роботи з моделями

AnyLogic – це чудовий інструмент для моделювання та оптимізації складних систем. Програму можна застосувати у багатьох сферах роботи. Вона є досить гнучкою та універсальною. Але через високу ціну та вимоги до системи, вона може бути менш доступною для користувачів

Розділ 2. ПРОЕКТ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1. Моделювання процесів прийняття рішень у конфліктних ситуаціях. Аналіз конфліктних ситуацій

Аналіз конфліктних ситуацій – це ключовий етап у моделювання процесу прийняття рішень на основі теорії ігор [11]. Для того аби максимально ефективно розв’язувати такі ситуації, потрібно в першу чергу визначити усі їхні характеристики, класифікувати конфлікти за певними критеріями та формалізувати їх.

В даному розділі розглянемо загальну характеристику конфліктів, їх структура та особливості. Також опишемо методологію формалізації задач, що дозволяє створити з конфліктної ситуації математичну модель, для того щоб далі аналізувати та розв’язувати за допомогою алгоритмів.

2.1.1. Характеристика конфліктів

Конфліктна ситуація являє собою взаємодію двох або більше суб’єктів (гравців). Кожен гравець має свої власні інтереси та протягом гри намагатиметься максимізувати власну вигоду. В таких ситуаціях дії одного гравця можуть впливати на виграші інших. Така залежність між усіма гравцями створює потребу в аналізі стратегій та прийняття оптимальних рішень.

Конфліктні ситуації мають такі характеристики [18]:

1. Наявність сторін конфлікту – грають два або більше суб’єкта, кожен з яких має свої цілі
2. Взаємозалежність рішень – результати гравців залежать як від їхніх дій, так і від дій противників
3. Протиріччя інтересів – зазвичай цілі гравців частково суперечать одна одній
4. Обмежені ресурси – зазвичай у конфліктах буває дефіцит ресурсів, таких як гроші, час

5. Наявність стратегій та виграшів – у кожного з гравців є свої стратегії. Вони допомагають досягти поставлених цілей, або мінімізувати витрати при програші

Конфлікти також мають декілька основних типів [18]:

1. Конкурентні – всі гравці діють у власних інтересах і намагається максимізувати виграш за рахунок інших.

2. Кооперативні – гравці співпрацюють між собою, задля отримання спільної вигоди.

3. Змішані – це ігри, в яких гравці можуть грати самі по собі, або кооперуватись.

Розуміння всіх характеристик конфліктів дозволяє коректно формалізувати задачі та перейти до математичних моделей для її розв'язання.

2.1.2. Метод формалізації задач

Одним із найважливіших етапів для аналізу конфліктних ситуацій у теорії ігор це формалізація [11]. Вона дозволяє сформувати взаємодії гравців у математичну модель, яка може бути розв'язана за допомогою відповідних алгоритмів та методів. Формалізацію задачі можна поділити на декілька етапів:

1. Визначення гравців – ідентифікація всіх учасників та їхніх можливих стратегій.

2. Побудова множини стратегії – опис можливих варіантів поведінки кожного гравця.

3. Формування матриці виграшів – визначення виграшів та втрат для кожного з гравців залежно від їхніх стратегій.

4. Тип гри – встановлення характеристик гри.

5. Вибір методу розв'язку – застосування певного підходу для аналізу ситуацій.

Один із найпоширеніших підходів для формалізації є використання матричних ігор [11]. В даному підході взаємодії всіх гравців представлені у

форматі матриці виграшів. Вона дозволяє визначати найоптимальніші стратегії за допомогою математичних методів.

Також не менш популярним є графічне представлення конфлікту. Він застосовується у динамічних іграх, де рішення приймаються послідовно. Усі результати візуалізуються у вигляді дерева рішень.

Формалізація задачі є необхідною умовою для подальшого використання алгоритмі оптимізації та аналізу стратегій, що будуть розглянуті в наступних розділах.

2.2. Постановка моделі для конкретної конфліктної ситуації

Одним із ключових етапів застосування теорії ігор для моделювання процесу прийняття рішень є формалізація конкретної конфліктної ситуації у вигляді математичної моделі. Такий підхід дозволяє відійти від абстрактного аналізу та зосередитись на практичному аспекті конфлікту між реальними суб'єктами. Під час даного етапу потрібно:

- Чітко окреслити гравців
- Визначити їхні цілі
- Виставити допустимі обмеження
- Створити перелік можливих стратегій поведінки
- Описати наслідки кожної комбінації стратегії у вигляді виграшів для кожного з гравців

Вибір кожної моделі залежить від характеру їхньої взаємодії, типу конфлікту та всієї доступної інформації. У випадку з двосторонньою взаємодією з чітко визначеним набором стратегій та результатів доцільно застосовувати матричну модель гри. Вона дозволить наочно представити взаємозв'язок між стратегіями та результатами. Побудова такої моделі потребує глибокого розуміння природи конфлікту.

В даному пункті буде розглянуто формування прикладної моделі гри для конфліктної ситуації, яка відповідає темі дослідження. Буде здійснено вибір

учасників гри, формалізація цілей, обмеження та обґрунтовано набір стратегій та побудована відповідна матриця виграшів. Це закладе підґрунтя для подальшого аналізу гри, зокрема визначення домінантних стратегій та рівноваги Неша.

Вибір учасників, визначення їхніх цілей та обмежень.

Вибір учасників це дуже важливий етап у формалізації конфліктної ситуації. Гравці – це суб'єкти, які приймають рішення умовах конфлікту. При цьому їхні рішення можуть впливати на результат усього процесу. У контексті теорії ігор гравці можуть бути фізичними особами, організаціями, державами або іншими структурами, які мають свої інтереси та визначені цілі. У нашому випадку, оскільки мова йде про застосування теорії ігор до конфліктних ситуацій, ми маємо справу із суб'єктами, які змагаються між собою за ресурси, вплив чи інші цілі.

2.2.1. Визначення гравців

В рамках нашої моделі ми будемо розглядати двох гравців, хоча в типових конфліктних ситуаціях кількість може бути різною.

Наприклад:

- Гравець А – одна із сторін конфлікту, яка розглядається як активний учасник з метою досягнення вигоди
- Гравець В: інший учасник конфлікту, який має свої цілі та може впливати на дії першого гравця.

В даному випадку як приклад ми можемо розглянути дві компанії, які конкурують між собою на ринку. Кожна з них повинна вибрати одну з стратегій: зниження цін чи збереження поточних цін.

2.2.2. Визначення цілей гравців

Цілі гравців у теорії ігор — це ті результати, яких прагне досягти кожна сторона конфлікту в процесі прийняття рішень. У контексті конфліктної ситуації кожен гравець діє раціонально, обираючи ту стратегію, яка дозволить досягти найкращого можливого для нього результату. Таким чином, постановка цілей —

це ключовий крок для побудови ігрової моделі, що відображає реальну поведінку суб'єктів у конфлікті.

Цілі можуть бути різними за природою:

- Економічні (прибуток, зниження витрат, збільшення ринкової частки);
- Політичні (посилення впливу, нейтралізація конкурента);
- Соціальні (отримання підтримки, підвищення авторитету);
- Технічні (оптимізація ресурсів, мінімізація часу виконання).

У нашому прикладному випадку розглядається конфлікт між двома фірмами, що працюють на одному ринку. Кожна з фірм — гравець — має свою ціль, яка формує основу для вибору стратегічних рішень.

Ціль гравця А (Компанія А). Компанія А прагне **максимізувати свій прибуток**, збільшивши обсяг продажів або частку ринку. Її рішення будуть базуватись на співвідношенні між витратами, доходами та очікуваною поведінкою конкурента.

Можливі підцілі:

- Отримання переваги в ціноутворенні — компанія може розглядати стратегію зниження цін для залучення клієнтів конкурентної компанії.
- Збереження або розширення частки ринку — важливо не допустити втрату клієнтів через дії конкурента.
- Зменшення втрат у випадку конфліктного сценарію — якщо конкуренція стає агресивною, компанія може обрати обережну стратегію для мінімізації втрат.

Раціональне міркування: компанія А діє у межах певного бюджету і прагне забезпечити вигідне співвідношення між витратами на маркетинг, знижки тощо, і потенційною вигодою, яку можна отримати через зміни у поведінці.

Ціль гравця В (Компанія В). Компанія В має аналогічну головну мету — **отримати максимальний прибуток**, але може мати інші пріоритети через іншу позицію на ринку, інші ресурси або стратегічні підходи.

Можливі підцілі:

- Утримання стабільного прибутку — компанія В може не прагнути агресивного розширення, а намагатися стабілізувати свої доходи.
- Захист лояльної клієнтської бази — замість зниження цін, фірма може інвестувати в покращення обслуговування або якість товару.
- Довгострокове позиціонування — компанія може обирати консервативні стратегії, щоб не зіпсувати репутацію на ринку через агресивну політику.

Раціональне міркування: компанія В, можливо, уже має усталену репутацію та клієнтську базу, тож її стратегія може бути націлена на уникнення короткострокових втрат заради довгострокової стабільності.

Взаємозалежність цілей гравців. Важливо підкреслити, що в умовах конфлікту цілі гравців не є незалежними: **успіх одного часто або опосередковано означає втрати для іншого** (особливо у випадках ігор з нульовою або фіксованою сумою). Таким чином, гравці не тільки визначають власні цілі, але й будують свої стратегії, враховуючи можливі дії іншої сторони.

Формалізація цілей. Цілі гравців у теорії ігор формалізуються за допомогою функцій виграшу (payoff functions), які обчислюють результат (прибуток, втрата тощо) кожного гравця в залежності від обраної комбінації стратегій.

- Функція виграшу гравця А:

$U_A(s_A, s_B)$ — виграш А за умов вибору стратегії s_A проти стратегії s_B гравцем В.

- Функція виграшу гравця В:

$U_B(s_A, s_B)$ — виграш В при тих же умовах.

Ці функції ляжуть в основу **матриці виграшів**, яку буде побудовано у наступному підпункті. Вони дозволяють об'єктивно порівнювати наслідки різних стратегій та знаходити рівноважні ситуації — зокрема, рівновагу Неша або домінантні стратегії.

2.2.3. Обмеження гравців

У теорії ігор **обмеження гравців** — це ті умови, правила або ресурси, які обмежують вибір стратегій для кожного учасника гри [11]. Вони безпосередньо впливають на можливість досягнення поставлених цілей і суттєво визначають характер поведінки в конфліктній ситуації.

Обмеження не є просто технічними деталями — вони визначають **реалістичність** моделі та дозволяють уникнути припущень, які б суперечили реальній ситуації. Включення обмежень у модель робить її ближчою до практичного застосування, зокрема в економіці, політиці чи управлінні.

Типи обмежень, які враховуються у моделі:

1. Ресурсні обмеження

Кожен гравець має обмежені ресурси для реалізації своїх стратегій:

- Обмеження бюджету (на рекламу, інновації, ціноутворення тощо).
- Обмеження виробничих потужностей.
- Обмеження у людських ресурсах або часі.

Приклад: компанія А не може дозволити собі одночасно знизити ціни і запустити нову рекламну кампанію через обмежений маркетинговий бюджет.

2. Інформаційні обмеження

Часто гравці не мають повної інформації про ситуацію:

- Вони можуть не знати точних витрат або стратегій опонента.
- Можуть мати обмежений доступ до ринкової аналітики.

Приклад: компанія В не знає, які саме продукти буде просувати конкурент, тому ухвалює рішення з огляду на найгірший сценарій.

3. Правові та регуляторні обмеження

Деякі дії гравців можуть бути обмежені законодавчо:

- Антимонопольне законодавство.
- Заборона на певні форми цінової конкуренції.
- Етичні або соціальні норми поведінки.

● **Приклад:** фірмі заборонено демпінгувати, тому вона не може вибрати стратегію занадто низької ціни.

4. Стратегічні обмеження

Внутрішні політики компанії або стратегічна позиція також можуть слугувати обмеженням:

- Відмова від агресивного просування на шкоду довгостроковій репутації.
- Прагнення зберегти стабільність замість ризикованого розширення.

Приклад: компанія А хоче уникати надмірного зниження цін, щоб не знецінити бренд.

5. Обмеження за кількістю стратегій

У математичному моделюванні часто встановлюється скінченна кількість допустимих стратегій, щоб зробити гру розв'язуваною.

Приклад: кожна компанія може обрати лише одну з трьох стратегій: «низька ціна», «висока якість», «агресивна реклама».

Формалізація обмежень у моделі [11]

У рамках математичної постановки задачі обмеження зазвичай враховуються як:

- **Обмеження на допустимі стратегії:** множина стратегій кожного гравця обмежується певними умовами $S_i = \{s_{i1}, s_{i2}, \dots, s_{in}\}$.
- **Функціональні обмеження:** у моделі можуть бути представлені у вигляді нерівностей або логічних умов, наприклад:

$$C_A(s) \leq B_A \text{ (вартість стратегії не перевищує бюджет гравця А)}$$

Вплив обмежень на поведінку гравців

Обмеження формують рамки раціональної поведінки гравців. Гравець, який не врахує власні обмеження, може обрати стратегію, яка здається виграшною, але в реальності її неможливо реалізувати. Таким чином, ефективна стратегія — це не просто оптимальна з погляду виграшу, а **реально досяжна** з урахуванням обмежень.

2.2.4. Визначення стратегій для гравців

Стратегія — це повний план дій гравця на кожен можливий випадок розвитку подій. Формування множини стратегій дозволяє перейти від загального опису конфлікту до його формалізованої моделі, наприклад, у вигляді матриці виграшів.

Типи стратегій у моделюванні

1. Чисті стратегії

Це конкретні, однозначно визначені дії, які гравець обирає у грі. Наприклад:

- Встановити високу/низьку ціну на товар.
- Обрати агресивну/помірну маркетингову політику.
- Вийти/не вийти на новий ринок.

У випадку **скінченних ігор** кожен гравець має скінченну множину таких стратегій:

$$S_A = \{s_{A1}, s_{A2}, \dots, s_{Am}\}, S_B = \{s_{B1}, s_{B2}, \dots, s_{Bn}\}$$

2. Змішані стратегії

Це імовірнісний вибір між кількома чистими стратегіями. Наприклад, гравець А може з імовірністю 60% обрати одну стратегію, а з 40% — іншу.

Такі стратегії часто використовуються для аналізу рівноваги Неша, коли жодна чиста стратегія не є однозначно вигідною.

3. Домінантні стратегії

Це стратегії, які забезпечують найкращий результат незалежно від вибору противника. Якщо така стратегія існує, її обирають у першу чергу.

При моделюванні конкретної ситуації — наприклад, конкуренції двох фірм — набір стратегій можна визначити з аналізу їхніх можливостей і цілей.

Приклад:

· Гравець А (компанія А):

- Стратегії:

s_{A1} : зниження ціни,

s_{A2} : посилення реклами,

s_{A3} : впровадження нової технології.

· Гравець В (компанія В):

- Стратегії:

S_{B1} : зниження ціни,

S_{B2} : посилення реклами,

S_{B3} : впровадження нових технологій.

Кожна з цих стратегій повинна відповідати:

- реальним можливостям гравця (з урахуванням ресурсних обмежень);
- логіці досягнення поставлених цілей;
- реакції на можливу поведінку супротивника.

Зі збільшенням кількості можливих стратегій складність гри зростає експоненційно, тому в реальному моделюванні часто виконують попередній аналіз стратегій:

- Відкидають явно невігідні.
- Об'єднують схожі.
- Виділяють основні лінії поведінки.

Таким чином, визначення стратегій — це не лише технічний етап, а аналітичний процес, що вимагає врахування контексту, обмежень і цілей.

2.2.5. Формування матриці виграшів

Після визначення гравців, їхніх стратегій, цілей та обмежень, наступним кроком у побудові моделі конфліктної ситуації є формування матриці виграшів. Це ключовий елемент моделювання у теорії ігор. Він дозволяє формалізувати ситуацію у вигляді таблиці і застосувати математичні методи аналізу.

2.2.6. Сутність матриці виграшів

Матриця виграшів – це двовимірна таблиця, в якій рядки відповідають стратегіям одного з гравців, а стовбці – стратегіям іншого гравця. Елемент матриці a_{ij} містить виграш або втрату гравця А за умов що він обрав стратегію S_{Ai} , а гравець В – стратегію S_{Bj} . [11]

У іграх з нульовою сумою виграш одного з гравців дорівнює програшу другого.

2.3. Процедура формування

2.3.1. *Визначення всіх можливих пар стратегій*

Для кожного поєднання стратегій визначається результат взаємодії – виграш або втрата. Це може бути: грошовий прибуток, частина ринку, кількість клієнтів, тощо.

2.3.2. *Оцінка результатів взаємодії*

Результати можуть бути задані:

- Експертно – за допомогою фахових оцінок
- Історично – на прикладі схожих ситуацій в минулому
- Аналітично – через формули чи моделі, які будуть враховувати ресурси, витрати, попит, тощо [20]

2.3.3. *Заповнення матриці*

Виграші вписуються в комірки матриці відповідно до конкретних стратегій

Розглянемо приклад:

	B1: Знижки	B2: Реклама
A1: Знижки	(2, -2)	(5, -5)
A2: Реклама	(3, -3)	(4, -4)

У матриці значення в дужках – виграші гравця А у грі з нульовою сумою. Для більш загального випадку матриця може містити пари виграшів для обох гравців.

Матриці виграшів стають основою для:

Пошуку домінантних стратегій

- Побудови рівноваги Неша

- Застосування алгоритмів оптимізації
- Імітаційного або комп'ютерного моделювання

2.3.4. Формування матриці виграшів для конкретної ситуації – Конфлікт між двома компаніями конкурентами:

У межах поставленої задачі моделювання конфліктної ситуації було обрано сценарій взаємодії двох компаній-конкурентів, які діють на одному ринку. Ці компанії — гравці у грі — прагнуть зміцнити свої ринкові позиції, залучити більшу кількість клієнтів, і при цьому мають обмежені ресурси для реалізації маркетингових стратегій.

Визначення стратегій гравців:

Компанія А та компанія В мають однаковий набір із трьох стратегій:

- S1 – Зниження цін на продукцію
- S2 – Посилення реклами
- S3 – впровадження нової технології

Кожна стратегія має свої сильні та слабкі сторони. Наприклад, зниження цін — агресивна короткострокова тактика, реклама — спосіб збільшення впізнаваності, а технології — інвестиція в довгострокову якість і ефективність.

Принцип побудови матриці.

Матриця будується за принципом: на перетині стратегій гравців фіксується виграш для кожного з них, обрахований у вигляді зміни частки ринку. Виграші умовні (в балах, які відповідають експертній оцінці ефективності стратегій в умовах конкуренції). Для простоти подавання використовують гру з нульовою сумою.[11]

Матриця виграшів:

	В: Зниження цін	В: Реклама	В: Технології
А: Зниження цін	(0,0)	(1, -1)	(2, -2)
А: Реклама	(-1, 1)	(0,0)	(2, -2)

А: Технології	(-2, 2)	(-2,2)	(0,0)
---------------	---------	--------	-------

Пояснення до матриці:

- (0, 0) — стратегія обох компаній однакова, відбувається баланс сил.
- (1, -1) — реклама поступається зниженню цін, адже покупець іде за вартістю.
- (2, -2) — інновації виявляються виграшними проти старих підходів.
- (-2, 2) — інвестиції в технології не приносять негайного ефекту проти агресивних стратегій суперника.

Ця матриця ілюструє, як вибір стратегії залежить не лише від внутрішніх ресурсів компанії, а й від передбачення дій суперника, що характерно для конфліктного прийняття рішень.

2.4. Застосування рівноваги Неша та інших підходів для пошуку оптимального рішення

Рівновага Неша— один з основоположних концептів теорії ігор, що дозволяє знайти стійке рішення, за якого жоден з гравців не має стимулу змінювати свою стратегію в односторонньому порядку, якщо стратегія іншого залишається незмінною [19].

Проаналізуємо нашу матрицю виграшів:

	В: Зниження цін	В: Реклама	В: Технології
А: Зниження цін	(0,0)	(1, -1)	(2, -2)
А: Реклама	(-1, 1)	(0,0)	(2, -2)
А: Технології	(-2, 2)	(-2,2)	(0,0)

Крок 1. Пошук найкращої відповіді гравця А

Для кожної фіксованої стратегії В обираємо максимальні виграші А:

- Якщо В грає S1 (Зниження цін) → А обирає S1 (виграш 0)
- Якщо В грає S2 (Реклама) → А обирає S1 (виграш 1)
- Якщо В грає S3 (Технології) → А обирає S1 або S2 (виграш 2)

Крок 2. Пошук найкращої відповіді гравця В

Для кожної фіксованої стратегії А обираємо максимальні виграші В:

- Якщо А грає S1 (Зниження цін) → В обирає S1 (виграш 0)
- Якщо А грає S2 (Реклама) → В обирає S1 (виграш 1)
- Якщо А грає S3 (Технології) → В обирає S1 або S2 (виграш 2)

Висновок:

Перетин найкращих відповідей для обох гравців - Стратегія (S1, S1) → (0, 0) — єдина чиста рівновага Неша.

Це означає, що обидві компанії обирають стратегію зниження цін, що створює взаємну рівновагу: жодна компанія не може покращити свій виграш, змінивши стратегію в односторонньому порядку.

Цей результат демонструє класичний "ціновий конфлікт" на ринку. Хоча стратегія впровадження нових технологій або посилення реклами може дати перевагу в окремих випадках, у стабільному стані обидва гравці надають перевагу зниженню цін, побоюючись втратити ринкову частку. Це приклад неоптимального, але стійкого рішення, що в реальній економіці трапляється часто.

Модель гри можна представити у вигляді дерева, а саме на рисунку 2.1:

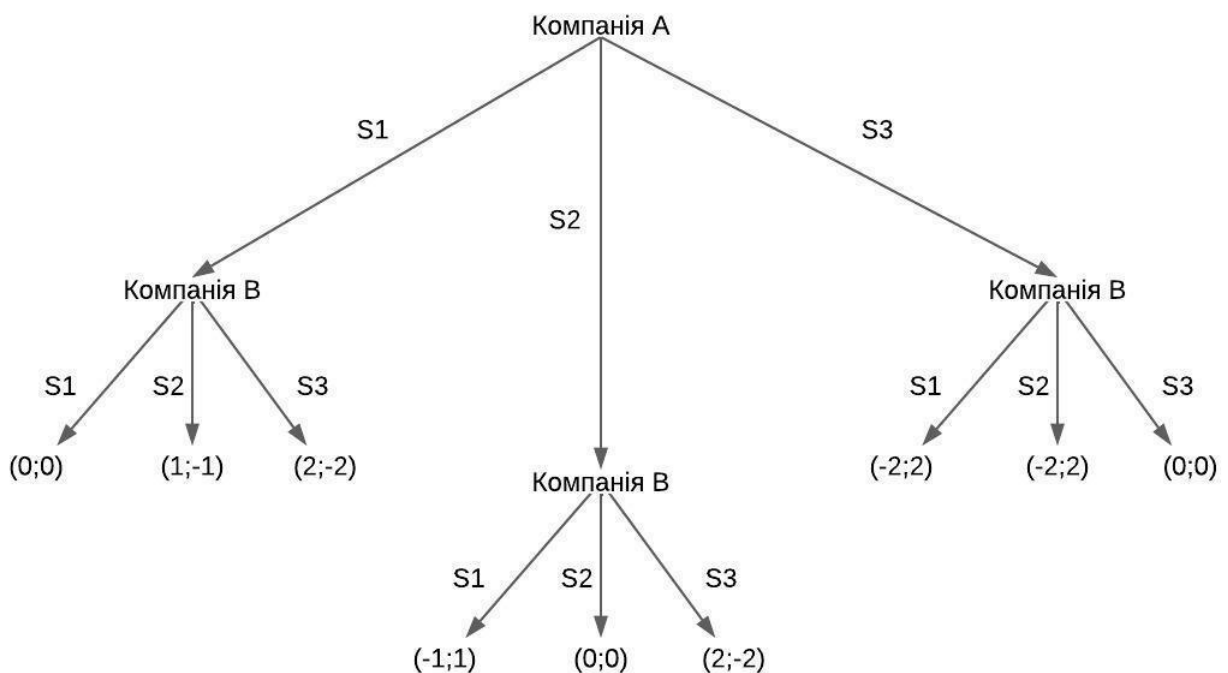


Рисунок 2.1. - Модель гри у вигляді дерева

2.5. Застосування методу домінування

Метод домінування — це базовий інструмент у теорії ігор, який дозволяє спростити матрицю виграшів шляхом виключення домінованих стратегій, тобто таких, які ніколи не дають кращого результату, ніж інша стратегія незалежно від вибору суперника. [21]

Проведемо аналіз нашої матриці

Крок 1. Пошук домінованих стратегій для гравця А:

Порівнюємо рядки (виграші гравця А):

- A1 vs A2:
 $(0 > -1), (1 > 0), (2 = 2) \rightarrow$ A1 строго домінує A2
 \Rightarrow Стратегію A2 (Реклама) можна виключити.
- A1 vs A3:
 $(0 > -2), (1 > -2), (2 > 0) \rightarrow$ A1 строго домінує A3
 \Rightarrow A3 також виключаємо.

Отже, у гравця А залишається лише стратегія Зниження цін.

Крок 2. Пошук домінованих стратегій для гравця В:

Порівнюємо стовпці (виграші гравця В):

- B1 vs B2:
 $(0 > -1), (1 > 0), (2 > -2) \rightarrow$ B1 строго домінує B2
 \Rightarrow B2 виключається.
- B1 vs B3:
 $(0 > -2), (1 > -2), (2 > 0) \rightarrow$ B1 строго домінує B3
 \Rightarrow B3 також виключається.

У гравця В також залишається тільки Зниження цін.

Знову приходимо до єдиної стратегії (Зниження цін, Зниження цін) з виграшем $(0, 0)$.

Висновок:

Метод домінування, як і рівновага Неша, підтверджує, що стратегія зниження цін є оптимальною з точки зору стійкості, хоч і не обов'язково з погляду максимізації виграшу.

2.6. Аналіз гри у змішаних стратегіях

У реальних конфліктних ситуаціях учасники часто не обирають одну фіксовану (чисту) стратегію, а випадковим чином обирають між кількома можливими стратегіями згідно з певним розподілом імовірностей. Такий підхід називається змішаними стратегіями. [11]

Основна ідея

- Змішана стратегія — це набір ймовірностей, з якими гравець обирає свої чисті стратегії.
- Мета — максимізувати очікуваний виграш, враховуючи змішані стратегії опонента.

Формалізація:

Позначимо:

- p_1, p_2, p_3 — імовірності вибору стратегій A1, A2, A3 гравцем А (компанія 1)
- q_1, q_2, q_3 — імовірності вибору стратегій B1, B2, B3 гравцем В (компанія 2)

Матриця виграшів для гравця А:

	В: Зниження цін	В: Реклама	В: Технології
А: Зниження цін	0	1	2
А: Реклама	-1	0	2
А: Технології	-2	-2	0

Алгоритм розв'язку буде виглядати так:

- 1 Ввести матрицю виграшів А.
- 2 Побудувати обмеження у вигляді системи нерівностей.
- 3 Побудувати об'єктивну функцію — максимізацію v .

4 Сформулювати задачу як задачу лінійного програмування.

5 Розв'язати її чисельно (наприклад, симплекс-методом або за допомогою бібліотеки).

6 Вивести отримані значення p_1, p_2, p_3 — тобто оптимальну змішану стратегію.

Пізніше ми реалізуємо це за допомогою програми.

У даному підрозділі було розглянуто кілька підходів до визначення оптимальних стратегій у конфліктних ситуаціях: рівновага Неша, домінантні стратегії, сідлова точка та аналіз змішаних стратегій. Кожен із методів має свої переваги та застосовується залежно від типу гри та структури виграшів. Практична цінність таких підходів полягає в можливості обґрунтованого вибору дій у конкурентному середовищі. На основі розроблених алгоритмів можливе програмне моделювання та автоматизація аналізу конфліктних ситуацій, що й буде реалізовано в наступних розділах.

2.7. Розробка програмної моделі для аналізу конфлікту двох компаній

У цьому розділі представлено концептуальний опис програмної моделі, яка реалізує прикладне вирішення конфліктної ситуації на основі методів теорії ігор. Розроблене рішення дозволяє користувачу змоделювати суперечку між двома учасниками, вибрати стратегії, побудувати матрицю виграшів та знайти оптимальні стратегії за допомогою класичних підходів, зокрема рівноваги Неша та аналізу змішаних стратегій.

2.7.1. Вхідні дані

У цьому підпункті розглядається опис та структуру вхідних даних, які користувач надає програмі для проведення аналізу конфліктної ситуації. Всі введені дані повинні бути представлені у формі, зручній для подальшого обчислення та моделювання. Основні елементи, що складають вхідні дані:

1) Кількість гравців

Для аналізу конфлікту двох компаній, програма приймає вхідне значення, яке вказує кількість учасників гри. У цьому випадку — це два гравці (дві компанії), але програма повинна бути розроблена так, щоб у майбутньому легко адаптувати її для більшої кількості гравців.

2) Назви та опис гравців

Кожен гравець має бути ідентифікований у системі, тому користувач повинен вказати:

- Імена або назви компаній (наприклад, "Компанія А" та "Компанія Б").
- Короткий опис ролі кожної компанії, якщо це необхідно для

подальшого аналізу.

Ці дані допомагають ідентифікувати учасників гри при подальшому виведенні результатів і аналізі.

3) Цілі гравців

• Кожен з двох гравців (компаній) має певні цілі, які вони намагаються досягти через вибір стратегій.

• Вони можуть бути представлені у вигляді списків рядків. У програмі ці дані будуть збережені у змінних типу **List[str]**

4) Стратегії гравців

• Кожен гравець має три стратегії на вибір: зниження цін, посилення реклами та впровадження нової технології.

• Ці стратегії зберігаються у вигляді списків, де кожен елемент є стратегією для одного з гравців.

5) Матриці виграшів

• Матриця виграшів відображає виграші кожного гравця при виборі певних стратегій.

• Це двовимірний список, де кожен елемент є парою виграшів для двох гравців (А і В).

6) Ймовірності для змішаних стратегій

Для змішаних стратегій кожен гравець може вибирати стратегії не з певністю, а з ймовірністю. Ці ймовірності зберігаються як списки чисел типу float, що представляють ймовірності для кожної стратегії гравця.

2.7.2. Алгоритм вибору програмної реалізації

1) Обчислення виграшів для кожної можливої пари стратегій

Програма перебирає всі можливі варіанти стратегій для двох гравців, розраховуючи виграші для кожної пари стратегій з матриці виграшів. Для кожної пари стратегій A і B програма обчислює результат, помножуючи виграші на ймовірності вибору кожної стратегії.

2) Пошук рівноваги Неша

Пошук рівноваги Неша визначається за допомогою порівняння виграшів кожного гравця в кожній можливій стратегії. Якщо жоден з гравців не може покращити свій виграш, змінюючи стратегію, коли інший гравець зберігає свою стратегію, то така пара стратегій є рівновагою Неша. [22]

3) Розрахунок змішаних стратегій

Враховуючи ймовірності для кожної стратегії, програма обчислює середні виграші для кожного гравця, враховуючи не тільки чисті стратегії, але й змішані, де кожен гравець обирає стратегію з певною ймовірністю.

2.7.3. Інтерфейс користувача

1) Вибір стратегій

Користувач вибирає стратегію для кожного гравця (компанії) з трьох можливих варіантів: зниження цін, посилення реклами, впровадження нової технології. Цей вибір здійснюється через інтерфейс консольного введення.

2) Введення ймовірностей змішаних стратегій

Користувач може ввести ймовірності для змішаних стратегій. Програма зчитує ці значення з консолі для подальшого обчислення виграшів.

Дана програма надає можливість користувачеві оцінити різні стратегії та знайти оптимальні рішення у конфліктних ситуаціях між двома компаніями, застосовуючи основні методи теорії ігор, зокрема рівновагу Неша та змішані стратегії.

2.7.4. Аналіз результатів моделювання.

У цьому розділі представлено аналітичний огляд результатів, отриманих у процесі моделювання конфліктної ситуації між двома компаніями з використанням інструментів теорії ігор. Було досліджено, як вибір стратегій впливає на виграші обох учасників, а також визначено найвигідніші варіанти поведінки, що забезпечують стабільний результат за умов конкуренції.

У моделі, що була сформована раніше, кожен із гравців мав три доступні стратегії:

- **S1:** Зниження цін
- **S2:** Посилення реклами
- **S3:** Впровадження нової технології

Ці стратегії є типовими інструментами конкурентної боротьби на ринку. Вибір кожної зі стратегій мав прямий вплив на виграші обох компаній, що відображено у сформованій матриці виграшів. Кожен стовпчик і рядок цієї матриці відповідає поєднанню стратегій обох гравців, і таким чином, для кожної пари стратегій можна визначити результат у вигляді виграшу або втрати.

Аналізуючи матрицю, можна помітити:

Стратегія зниження цін (S1) призводить до короткострокового збільшення прибутку при незначному опорі з боку конкурента, але у довгостроковій перспективі може зменшити загальний дохід на ринку.

Посилення реклами (S2) ефективніше, коли конкурент обирає пасивні стратегії або не інвестує в інновації.

Впровадження нових технологій (S3) є високоризиковою, але може принести значні виграші при слабкому розвитку конкурента.

Було виявлено, що в умовах симетрії виграшів, обидва гравці мають мотивацію змінювати свої стратегії у відповідь на дії опонента. Саме тому необхідним було застосування аналізу рівноваги Неша, яка дозволила виявити точки стабільності — ситуації, у яких жоден з гравців не зацікавлений змінювати свою стратегію в односторонньому порядку.

2.7.5. Аналіз результатів моделювання.

У цьому розділі представлено аналітичний огляд результатів, отриманих у процесі моделювання конфліктної ситуації між двома компаніями з використанням інструментів теорії ігор. Було досліджено, як вибір стратегій впливає на виграші обох учасників, а також визначено найвигідніші варіанти поведінки, що забезпечують стабільний результат за умов конкуренції.

Вплив стратегій учасників на результат.

У моделі, що була сформована раніше, кожен із гравців мав три доступні стратегії:

- **S1:** Зниження цін
- **S2:** Посилення реклами
- **S3:** Впровадження нової технології

Ці стратегії є типовими інструментами конкурентної боротьби на ринку. Вибір кожної зі стратегій мав прямий вплив на виграші обох компаній, що відображено у сформованій матриці виграшів. Кожен стовпчик і рядок цієї матриці відповідає поєднанню стратегій обох гравців, і таким чином, для кожної пари стратегій можна визначити результат у вигляді виграшу або втрати.

Аналізуючи матрицю, можна помітити:

- Стратегія зниження цін (S1) призводить до короткострокового збільшення прибутку при незначному опорі з боку конкурента, але у довгостроковій перспективі може зменшити загальний дохід на ринку.

- Посилення реклами (S2) ефективніше, коли конкурент обирає пасивні стратегії або не інвестує в інновації.

- Впровадження нових технологій (S3) є високоризиковою, але може принести значні вигоди при слабкому розвитку конкурента.

Було виявлено, що в умовах симетрії вигащів, обидва гравці мають мотивацію змінювати свої стратегії у відповідь на дії опонента. Саме тому необхідним було застосування аналізу рівноваги Неша, яка дозволила виявити точки стабільності — ситуації, у яких жоден з гравців не зацікавлений змінювати свою стратегію в односторонньому порядку.

Виявлення вигідних стратегій.

Для визначення найкращих стратегій були застосовані такі підходи:

- Пошук доміантних стратегій: перевірка, чи існують такі стратегії, які завжди приносять вигідніший результат незалежно від вибору супротивника. У нашій моделі однозначно доміантних стратегій не було, що вказує на складність прийняття рішень у конкурентному середовищі.

- Рівновага Неша: при розрахунках змішаних стратегій (ймовірнісного розподілу вибору стратегій) вдалося знайти оптимальне поєднання, що задовольняє обидві сторони. Наприклад, для гравця А оптимальним є застосовувати стратегії з певними ймовірностями (p_1 , p_2 , p_3), що дозволяє знизити ризик і забезпечити стабільний результат у випадку повторюваних ігор.

- Критерій мінімаксу: допоміг у виборі найбільш обережної (консервативної) стратегії, що мінімізує максимально можливі втрати.

На основі аналізу можна зробити висновок, що змішані стратегії забезпечують більш стабільний результат у конфліктній ситуації, порівняно з чистими. Вони дозволяють адаптуватися до дій суперника, знижуючи ймовірність потрапити у програшну ситуацію.

Крім того, для обох компаній було рекомендовано уникати стратегій, які є ефективними лише за умови пасивності конкурента. Найвигіднішими є ті стратегії, які або відповідають точці рівноваги Неша, або мають найменші ризики за мінімаксним критерієм.

Розділ 3. КОНЦЕПЦІЯ РЕАЛІЗАЦІЇ ПРОГРАМНОГО ПРОДУКТУ

В цьому розділі буде представлено концепцію створення програмного засобу, який буде призначено для моделювання процесу прийняття рішень у конфліктних ситуаціях за допомогою теорії ігор. Даний програмний продукт дає можливість користувачеві вводити дані про учасників конфлікту, всі наявні стратегії, матрицю виграшів. Також можна буде здійснювати усі необхідні математичні операції для знаходження рівноваги Неша, домінантних стратегій і вигідних рішень у різних сценаріях.

Метою реалізації є створення функціонального інструменту, які можна використовувати у різних цілях, таких як навчання, практичні задачі, де потрібно буде проаналізувати дії гравців у певних конфліктних ситуаціях. Не менш важливим є створення простої і зрозумілої структури програми, яка забезпечить користувачам легкість взаємодії, а розробникам подальший розвиток та розширення програми.

У підпунктах даного розділу описані обрані інструменти та середовище програмування, структуру та архітектуру програми, всі обрані алгоритми, такі як алгоритми аналізу стратегій, формат введення та виведення даних, а також логіку взаємодії користувача з інтерфейсом програмного засобу.

3.1. Вибір інструментів розробки та обґрунтування

Щоб реалізувати даний продукт було обрано мову програмування Python. Ця мова є досить гнучкою, має великий спектр різноманітних бібліотек призначених для математичного моделювання, зручний синтаксис та наявність великої спільноти розробників.

Мета програми – моделювання конфліктних ситуацій між двома гравцями (компаніями) з можливістю аналізу усіх наявних стратегій, пошуку рівноваги Неша та вибору оптимальних рішень. Дана програма створена для вирішення

такої проблеми як допомога у прийнятті рішень у ситуаціях стратегічної конкуренції, де дії та результати усіх гравців взаємопов'язані.

Для програми було обрано теорію ігор, а саме матричні ігри для моделювання конфліктів з дискретним набором стратегій. У моделі враховуються ігри нульової суми, а також некооперативні ситуації, що є типовими для бізнес-середовища. Даний підхід дозволяє формалізувати поведінку гравців і надати чіткі алгоритмічні шляхи пошуку найкращих стратегій.

Програма розрахована як на навчальну аудиторію, так і на бізнес-аналітиків та стратегів, яким потрібно мати інструмент для швидкого аналізу рішень у конкурентних умовах.

Основні технології, які будуть використані:

- Python — основна мова розробки, обрана за її універсальність та легкість у реалізації алгоритмів;
- NumPy — для виконання математичних операцій та роботи з матрицями;
- Matplotlib або Plotly — для побудови графіків і візуалізації результатів;
- Tkinter або PyQt — для створення графічного інтерфейсу (на пізнішому етапі);
- Jupyter Notebook (на етапі тестування) — для інтерактивної розробки та демонстрації алгоритмів.

Використання усіх цих технологій допоможе створити зручний, ефективний та практичний програмний продукт. Python добре підходить для створення обчислювального модуля. У візуальній складовій нам допоможуть бібліотеки візуалізації. Вони забезпечать зрозуміле представлення результатів для користувачів.

3.2. Архітектура програмного продукту

Розроблена система моделювання конфліктних ситуацій на основі теорії ігор забезпечує користувачу низку важливих функціональних можливостей,

спрямованих на аналіз стратегій у грі з двома гравцями. Зокрема, користувач зможе:

- Вводити кількість гравців та їхні стратегії;
- Задавати матрицю виграшів для кожного гравця;
- Аналізувати домінантні стратегії;
- Визначати рівновагу Неша (у чистих та змішаних стратегіях);
- Отримувати візуалізацію результатів у вигляді таблиць та графіків;
- Отримувати текстові пояснення обчислень (для навчальних цілей);
- Зберігати або експортувати результати аналізу.

Архітектура програмного продукту спроектована таким чином, щоб забезпечити модульність, розширюваність та простоту продукту. Система розділена на декілька функціональних блоків, кожен з яких виконує свою роль: обробка даних, реалізація математичних моделей, візуалізація та взаємодія з користувачем [23]. Даний підхід відповідає принципам структурного та об'єктно-орієнтованого програмування.

Розглянемо основні компоненти системи (`input_module.py`):

1) Модуль введення даних

Цей модуль відповідає за отримання вхідних даних від користувача.

Дані включають:

- Кількість гравців (у нашому випадку — 2)
- Назви стратегій кожного гравця (**`strategies_A`**, **`strategies_B`**) – тип

`list[str]`

- Матрицю виграшів (**`payoff_matrix`**) – тип **`list[list[tuple[float, float]]]`**

2) Модуль обчислення (`calculation_module.py`):

У цьому модулі реалізуються:

- пошук домінантних стратегій;
- визначення рівноваги Неша;

- розв'язання ігри в змішаних стратегіях;
- виявлення оптимальних стратегій для кожного гравця;
- робота з матрицями виграшів.

Типові функції: **find_nash_equilibria()**, **solve_mixed_strategies()**.

3) Модуль візуалізації (`visualization_module.py`):

Здійснює побудову:

- таблиць зі стратегіями та виграшами;
- графіків змішаних стратегій (за допомогою Matplotlib або Plotly);
- схем конфлікту у вигляді дерева рішень (опційно).

4) Модуль інтерфейсу користувача (`console_interface.py`):

На початковому етапі передбачено консольний інтерфейс, у якому користувач:

- вводить свої дані або вибирає з прикладів;
- отримує результат у вигляді таблиць або коротких пояснень;
- може експортувати результати в `.txt` або `.csv`.

У перспективі можливе впровадження графічного інтерфейсу на базі Tkinter або PyQt.

5) Модуль збереження результатів (`export_module.py`):

Містить функції для експорту отриманих результатів аналізу (наприклад, **save_to_file()**), що дозволяє зберігати моделі конфліктів для подальшого аналізу або презентації.

Загальна архітектурна схема показана на рисунку 3.1:



Рисунок 3.1 – Загальна архітектурна схема

Як висновок можна сказати, що поділ програми на окремі модулі є досить зручним рішенням. Це дозволяє нам легко масштабувати та тестувати систему. Якщо нам потрібно буде внести якісь оновлення (нові типи ігор чи різні методи аналізу), достатньо буде лише додати або оновити вже існуючі модулі, уникаючи об'ємних змін в самій системі.

3.3. Опис функціональності програми

Даний програмний продукт розроблений для моделювання конфліктних ситуацій між двома гравцями (компаніями) з використанням методів теорії ігор. Основна мета – надати користувачеві інструмент для побудови, аналізу та візуалізації стратегічної взаємодії між сторонами з урахуванням заданих стратегій та можливих вигравів.

Основні функціональні можливості:

1. Створення моделі конфлікту:

- введення кількості гравців (фіксовано 2 на цьому етапі);
- визначення стратегій для кожного гравця (**strategies_A**, **strategies_B**);
- заповнення матриці вигравів (для кожної комбінації стратегій пара вигравів у форматі (**a_ij**, **b_ij**)).

2. Аналіз конфліктної ситуації:

- автоматичне визначення домінантних стратегій;
- пошук рівноваги Неша в чистих та змішаних стратегіях;
- аналіз оптимальних стратегій кожного гравця залежно від ситуації.

3. Робота з різними типами ігор:

- Антагоністичні ігри (ігри з нульовою сумою): виграти одного гравця прямо протилежні виграмам іншого.
- Некооперативні ігри з загальними виграшами: можливе існування ситуацій, вигідних для обох гравців.

- Матричні ігри: основний формат подання стратегій і вигравів.

4. Візуалізація результатів:

- побудова таблиць вигравів;
- графічне представлення результатів (у вигляді діаграм, дерев рішень тощо);
- інтерпретація стратегічних висновків.

5. Збереження та експорт результатів:

- збереження матриці та результатів аналізу в .txt або .csv;
- можливість повторного використання введених даних для серійного

моделювання.

6. Сценарії використання:

- Навчальні цілі — для студентів, які вивчають теорію ігор;
- Бізнес-аналітика — для оцінки ринкових стратегій конкурентів;
- Стратегічне планування — для прийняття обґрунтованих рішень в

умовах конфлікту інтересів.

Для прикладу, можна розглянути схему концептуального підходу теорії ігор у бізнесі. Схема охоплює декілька етапів. Першим етапом буде постановка задачі для кожної з компаній. Далі відбувається моделювання самого конфлікту. Потім кожна компанія обирає найвдалішу для себе стратегію. Після цього відбудеться аналіз результатів: якщо результат задовольняє кожну компанію, то гра закінчується. Якщо ні - гра продовжується. Весь алгоритм представлений в даній схемі на рисунку 3.2.

3.4. Модель даних та структура програми

Для ефективної реалізації продукту було розроблено чітку модель даних та структуровано логіку програми на основі принципів модульності та зрозумілої взаємодії між компонентами.

Назва змінної	Тип даних	Опис
strategies_A	list[str]	Список стратегій гравця А
strategies_B	list[str]	Список стратегій гравця В
payoff_matrix	list[list[tuple]]	Матриці виграшів у форматі [(A_win, B_win)]
dominant_strategies	dict[str, str]	Збереження домінантних стратегій для обох гравців
nash_equilibria	list[tuple[int, int]]	Позиції стратегій (i, j), які утворюють рівновагу Неша

mixed_strategies	dict[str, list[float]]	Оптимальні ймовірності змішаних стратегій
------------------	---------------------------	--

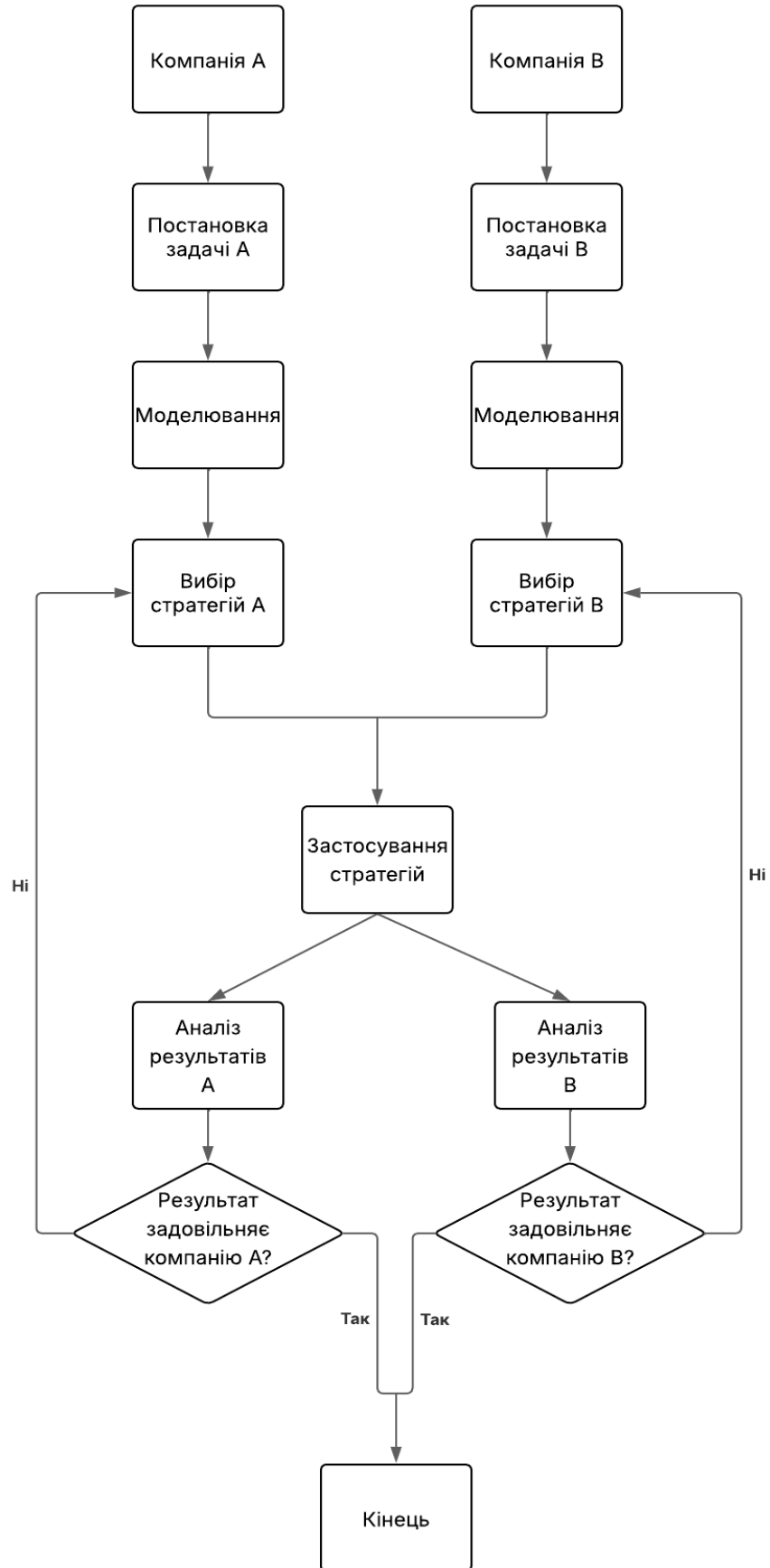


Рисунок 3.2 - Весь алгоритм

Основні об'єкти та типи даних [24]:

3.5. Приклад сценарію використання програми

Щоб продемонструвати функціональність програмного продукту, розглянемо типовий сценарій використання, який дозволяє змоделювати конфліктну ситуацію між двома компаніями та визначити оптимальні стратегії.

1) Запуск програми

Користувач відкриває програму (консольний застосунок або графічний інтерфейс — залежно від реалізації) і бачить запрошення до введення вихідних даних.

2) Введення даних

Програма запитує:

- Назви стратегій гравця А:

Введення прикладу:

Введіть 3 стратегії гравця А (через кому): Зниження цін, Посилення реклами, Впровадження нової технології

- Назви стратегій гравця В:

Введіть 3 стратегії гравця В (через кому): Зниження цін, Посилення реклами, Впровадження нової технології

- Матриця виграшів:

Користувач вводить комірки виграшів у форматі (виграш А, виграш В) для кожної комбінації стратегій. Наприклад:

Введіть виграші у форматі: (А,В)

Стратегія А1 проти В1: (3,3)

Стратегія А1 проти В2: (4,2)

3) Обробка даних

Після введення всіх даних, програма:

- формує матрицю виграшів;
- виконує пошук домінантних стратегій;

- визначає наявність рівноваги Неша;
- (опціонально) розраховує оптимальні змішані стратегії.

4) Вивід результатів

Програма виводить:

- Таблицю виграшів у зручному форматі;
- Виявлені домінуючі стратегії (якщо ϵ);
- Координати рівноваги Неша (наприклад: стратегія A2 проти B1);
- Рекомендації щодо вибору стратегії на основі аналізу.

5) Завершення або повторний запуск

Користувач може:

- Завершити програму;
- Повторити моделювання з новими даними;
- Експортувати результати у файл (у розширеній версії — у формат

CSV, PDF, PNG тощо).

Весь цей алгоритм можна представити у вигляді блок-схеми (рис.3.3).

Даний сценарій забезпечує зручну та послідовну взаємодію користувача з програмою. Також це дозволяє застосовувати їх як для навчальних так і для аналітичних бізнес-цілей

В цьому розділі було розглянуто та обґрунтовано концепцію створення програмного продукту для моделювання конфліктних ситуацій у грі з двома компаніями. Для розробки продукту було обрано таку мову програмування як Python та бібліотеки для обчислень і візуалізації, визначено архітектуру системи, що включає модуль введення даних, обчислювальний модуль, інтерфейс та модуль візуалізації.

Також було описано усі функціональні можливості програми та розглянуто сценарій її використання. Цей підхід дає змогу реалізувати ефективний інструмент як для навчальних так і прикладних цілей у сфері аналізу стратегічної взаємодії.

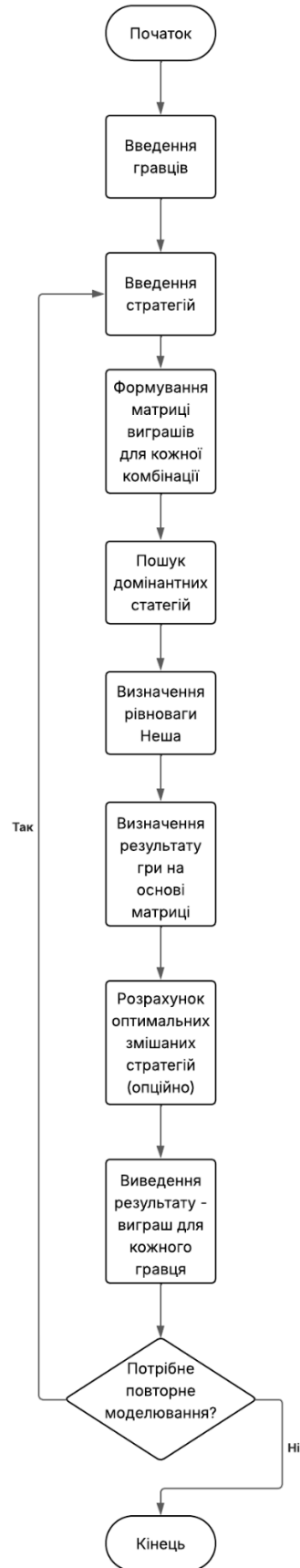


Рисунок 3.3 – Весь цей алгоритм у вигляді блок-схеми:

Розділ 4. ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШОГО РОЗВИТКУ

В даному розділі будуть підведені підсумки результатів проведеного дослідження, що були отримані під час розробки програмного продукту для моделювання конфліктних ситуацій із застосуванням теорії ігор. Було розглянуто приклад гри двох компаній, яким було потрібно максимізувати свій результат за допомогою певних стратегій. Це дозволило продемонструвати ефективність та зручність використанні математичних моделей у задачах стратегічного планування та прийняття рішень.

Також було проаналізована реалізація програмного забезпечення, його функціоналу та обмежень. Були поставлені задачі, які в майбутньому допоможуть вдосконалити та розвивати даний проект.

Даний розділ описує та оцінює те, наскільки досягнуті результати відповідають поставленій цілі, та вказує на потенційні шляхи розширення та адаптації моделі до складніших сценаріїв розвитку конфліктів.

4.1. Загальні висновки щодо результатів дослідження

Протягом усього процесу створення роботи було досягнуто основної задачі – створення програмного засобу для моделювання конфліктних ситуацій із застосуванням методів теорії ігор. Ми змогли побачити, що дана математична модель дозволяє формалізувати процес прийняття рішень у ситуаціях зіткнень інтересів обох сторін, що має широке практичне застосування як у навчальних моментах, так і економіці, менеджменту, маркетингу та політиці.

Для ситуації конфліктів двох компаній, які конкурують за ринок, було змодельовано набір певних стратегій, а саме зниження цін, посилення реклами та впровадження нових технологій. Було побудовано матрицю вигравів та на основі цієї матриці було знайдено рівновагу Неша – ключову точку, де жоден з гравців не мав стимулу змінювати свою стратегію в односторонньому порядку. Також було застосовано ще декілька методів аналізу, такі як виявлення домінантних стратегій

та аналіз змішаних стратегій. Це дозволило порівняти підходи до оптимізації рішень.

Було реалізовано програмний продукт на мові Python. Він має модульну архітектуру, зручний та зрозумілий для користувача інтерфейс введення даних та логічне структурування усіх алгоритмів. Усі логіка моделювання була ізольована у обчислювальному модулі. Це дозволяє зручно масштабувати програму та додавати нові функціональні можливості для нашого продукту.

Таким чином, результати дослідження засвідчили ефективність використання інструментів теорії ігор для аналізу конфліктних ситуацій. Змодельований приклад показав, як формалізований підхід може допомогти у виборі стратегічного рішення за умов невизначеності та взаємозалежності дій опонентів.

4.2. Оцінка функціональності реалізованого програмного продукту

На основі результатів тестування та аналізу функціоналу програми, можна зробити висновок, що створений продукт ефективно виконує поставлені завдання. Розглянемо декілька ключових моментів оцінки функціональних характеристик:

1) Наскільки добре працює програма?

Даний продукт стабільно працює в межах заявленого функціоналу. Він коректно обробляє вхідні дані, генерує матрицю виграшів для двох гравців, визначає домінуючі стратегії та знаходить рівновагу Неша (як у чистих, так і у змішаних стратегіях). Алгоритми працюють без критичних помилок, а результати збігаються з аналітичними обчисленнями, що свідчить про правильність реалізації логіки.

2) Повнота реалізованих функцій

Усі ключові функціональні можливості, заплановані на етапі проектування, реалізовані:

- Побудова та відображення матриці виграшів.
- Розрахунок рівноваги Неша.
- Визначення домінуючих стратегій.

- Підтримка аналізу змішаних стратегій.
- Вивід результатів у зручному для інтерпретації вигляді.

Отже, програмний продукт на даному етапі реалізує повноцінний цикл аналізу конфлікту між двома компаніями в рамках матричної гри.

3) Зручність користування

Інтерфейс програми структурований логічно. Це дає можливість користувачу вводити усі потрібні дані, отримувати результати аналізу та інтерпретувати їх. Відображення результатів супроводжується поясненнями, що полегшує сприйняття навіть для користувачів без поглибленої математичної підготовки.

У перспективі планується реалізація більш зручного детального графічного інтерфейсу.

4) Потенційна область застосунку

Програмний продукт може бути успішно використаний у таких сферах:

- Навчальні цілі – як практичний інструмент для вивчення теорії ігор, аналізу стратегій та рівноваги.
- Бізнес – для моделювання конкурентних ситуацій між компаніями, аналізу ринкової поведінки.
- Стратегічне планування – у випадках, коли необхідно оцінити поведінку опонента та обрати оптимальну реакцію в умовах конфлікту чи конкуренції.
- Дослідження в галузі прийняття рішень – для моделювання ситуацій, де важлива взаємодія між сторонами з протилежними або частково узгодженими інтересами.

Як висновок можемо сказати, що реалізований програмний продукт демонструє повну відповідність поставленим цілям. Він є ефективним інструментом для моделювання та аналізу конфліктних ситуацій у форматі гри двох гравців. Завдяки гнучкості архітектури та потенціалу до розширення, система має перспективу подальшого розвитку та широкого застосування в освітніх і практичних задачах.

4.3. Обмеження створеної моделі та програмного засобу

Не дивлячись на те, що програмний продукт реалізований досить успішно, можна побачити низку обмежень, які пов'язані з математичною моделлю, які є основою системи, та з функціональністю програмного забезпечення. В даному підпункті ми проаналізуємо основні обмеження, що можуть впливати на точність, масштабованість та універсальність використання розробленого продукту.

1) Обмеження математичної моделі

a. Двоосібна гра: дана реалізація продукту дає можливість моделювати конфлікт лише з двома гравцями. Сучасні ситуації часто передбачають участь трьох і більше сторін конфлікту. Для моделювання та розв'язку таких моделей потрібно використання більш складних моделей.

b. Матрична форма представлення: усі стратегії гравців описані у формі матриці виграшів. Це не дозволяє нам повноцінно змоделювати такі ситуації як ситуації з неповною інформацією, динамічними змінами або багатокроковими сценаріями.

c. Статичність гри: дана модель передбачає одноразовий вибір стратегії кожним з гравців. Це обмежує гнучкість моделювання ситуацій із неповною інформацією, динамічними змінами або багатокроковими сценаріями.

d. Нульова сума: у деяких реалізаціях припускається ситуація, коли виграш одного гравця дорівнює програшу іншого. У сучасних конфліктах частіше зустрічаються ігри з ненульовою сумою, де бувають взаємовигідні результати. Цей момент також потрібно врахувати та пропрацювати.

2) Обмеження програмного засобу

a. Тип інтерфейсу: дана реалізація має консольний інтерфейс. Це може бути незручними для кінцевих користувачів, які очікують наявність графічного інтерфейсу з кнопками, діаграмами та інтерактивними елементами.

b. Обмежена візуалізація: Візуалізація результатів обмежується базовими діаграмами. Не реалізовано повноцінну інтерактивну графіку або гнучку систему звітності.

с. Фіксована кількість стратегій: для кожного з гравців зафіксовано лише три стратегії. Щоб додати або забрати певні стратегії, користувач повинен змінювати програмний код вручну, що знижує зручність користування та універсальність.

d. Відсутність модулів машинного навчання або адаптивного аналізу: програма не враховує історичні дані або статистики попередніх рішень, що могло б покращити передбачення стратегій опонентів у реальному світі.

3) Обмеження з боку користувача

a. Необхідність базових знань: для ефективного користування продуктом, користувачу потрібно мати базові знання з теорії ігор. Звичайно програма розрахована на користувачів, які знаються на теорії ігор, але для нефахівців це може стати проблемою

b. Недостатній супровід: в даній версії не реалізовано повноцінну систему для пояснень, довідок та інструкцій для користувача. Це може ускладнити початок роботи з програмою

Поточна модель виконує свої основні функції – моделювання конфлікту між двома гравцями та аналіз їхніх стратегій. Але для розширення функціоналу продукту, збільшення кількості сфер можливого застосування та підвищення гнучкості системи, потрібно пропрацювати низку проблем та реалізувати багато моментів, таких як підтримка множинних гравців, динамічних сценаріїв, графічного інтерфейсу та адаптивних аналітичних можливостей. Усі ці напрямки можуть стати основою для перспективи подальшого розвитку програмного продукту. Ці моменти будуть розглянуті у наступному підпункті.

4.4. Перспективи подальшого розвитку програмного продукту

Поточна реалізація програмного продукту – це базовий прототип системи моделювання конфліктних ситуацій з двома гравцями. Вона дозволяє проводити аналіз простих матричних ігор, визначати рівновагу Неша, домінантні стратегії та досліджувати вплив стратегій на результати. Хоча якщо розглядати сучасні реальні конфлікти, то можна побачити, що функціоналу нашого продукту може

бути недостатньо, так як реальні конфлікти є складніші, маючи наприклад більше стратегій чи певні обмеження. Тому існує значний потенціал для подальшого розвитку і розширення функціоналу нашого програмного продукту.

Розглянемо декілька можливих перспективних напрямів для розвитку нашої програми:

1) Розширення кількості гравців

Поточна модель працює лише з двома гравцями, що є не зовсім реалістично для сучасних умов. Тому в майбутньому можна розширити кількість гравців до трьох та більше. Це наблизить модель до більш реальних умов конкуренцію у будь яких сферах, таких як бізнес, політика, усілякі стратегії.

2) Підтримка змішаних стратегій

Хоча теоретично підтримка змішаних стратегій описана, майбутні версії програми можуть включати алгоритмічне обчислення оптимальних змішаних стратегій, наприклад, через симплекс-метод або з використанням чисельних методів (метод лінійного програмування).

3) Покращення інтерфейсу для користувача

Наша програма розроблена як консольна. Графічний інтерфейс неабияк підвищить зручність роботи з продуктом. Користувачі зможуть легше взаємодіяти з програмою (переглядати матриці, будувати графіки, діаграми, тощо).

4) Інтеграція з базами даних

В майбутньому можлива інтеграція з базами даних. Це дозволить зберігати конфліктні сценарії, історії рішень, стратегії, шаблони конфліктів. Це дозволить повторно використовувати існуючі моделі та здійснювати повторний аналіз.

5) Візуалізація рішень

Розробка модуля візуалізації дозволить наочно представляти точки рівноваги, переваги стратегій, зміни у виграшах які залежатимуть від умов дій гравців. Цей модуль покращить сприйняття всієї інформації. Це особливо важливо у навчальному контексті.

6) Інтеграція з системами прийняття рішень

Можливе також включення програм до більших систем підтримки рішень (DSS), де вона виступатиме як модуль аналізу стратегічної взаємодії

7) Підтримка інших типів ігор

Наступні версії можуть охоплювати кооперативні ігри, ігри з неповною інформацією, еволюційні ігри тощо. Це дозволить адаптувати програму до нових сфер застосування.

Діаграма можливих покращень [25]:

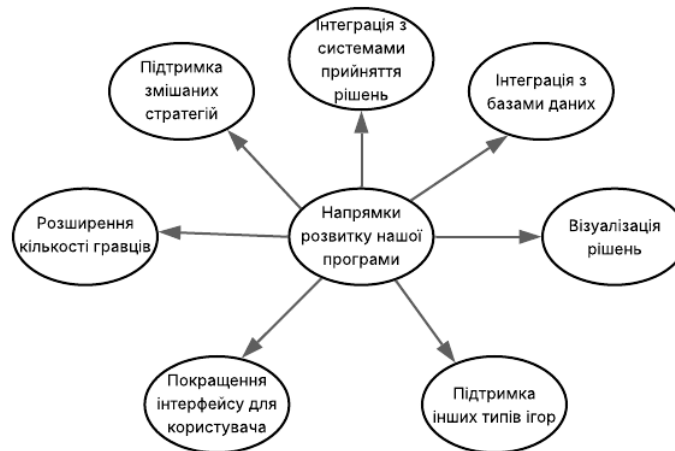


Рис.4.1. - Покращення програми

Отже ми можемо побачити, що майбутній розвиток нашого продукту має не лише технічні перспективи, а й методологічні. Його розвиток та покращення дозволить не лише використовувати систему у навчальних цілях, а й для розв'язку конфліктів у реальних умовах стратегічної взаємодії, удосконалюючи ефективність та обґрунтованість вибраних рішень.

ВИСНОВОК

У роботі було реалізовано комплексне дослідження процесу моделювання конфліктних ситуацій між двома компаніями з використанням теорії ігор. Основним аспектом була побудова математичної моделі стратегічної взаємодії, а також аналіз результатів моделювання на практичному прикладі.

Під час першого етапу ми сформулювали предмет дослідження та обґрунтували доцільність використання саме теорії ігор для вирішення задачі стратегічного аналізу. Було виявлено, що ситуації конкурентної боротьби між компаніями, в яких кожна сторона має обмежений набір стратегій, природно описуються у вигляді матричних ігор. Для моделювання було обрано тип гри з нульовою сумою. Даний тип гри дозволяє оцінити вигоди та втрати гравців у взаємозалежній системі.

Побудована модель охоплює три основних стратегії для обох гравців - зниження цін, посилення реклами та впровадження нових технологій. Спираючись на дані стратегії було створено матрицю виграшів, яка стала основою для подальшого аналізу. Щоб знайти найоптимальніші рішення для нашої задачі було застосовано кілька підходів: рівновага Неша, метод домінування, сідлова точка та аналіз змішаних стратегій. Застосування кожного з методів дозволяє детально охарактеризувати поведінку кожного з гравців та знайти найбільш вигідні стратегічні сценарії.

У другому етапі дослідження було розроблено програмний засіб на основі мови Python. У роботі повноцінно описано структури програми, включно з формалізацією вхідних даних, описом основних змінних, архітектури програми та застосовуваних технологій. Програма реалізує функціональність обчислення рівноваги Неша, аналізу домінуючих стратегій, побудови матриці виграшів та візуалізації результатів у вигляді діаграм. Розроблений інтерфейс та модуль обчислень дають змогу користувачу самому змоделювати ситуації стратегічного протистояння та знаходити оптимальні рішення.

Аналіз результатів моделювання показав, що не дивлячись на невелику кількість стратегій, можна виявити стабільні точки рівноваги, що відповідають

найкращим діям обох гравців. Було встановлено вплив кожної зі стратегій на результат гри та виявлено найбільш вигідні дії залежно від поведінки супротивника. Це підтверджує ефективність моделі для використання в навчальних цілях, а також для підтримки прийняття рішень у сфері бізнесу або стратегічного планування.

Не дивлячись на ефективність розробленої системи, вона також має свої обмеження. Проаналізувавши усю модель, були виявлені такі недоліки, як фіксована кількість гравців, обмежений набір стратегій, базова візуалізація та відсутність кооперативних сценаріїв. Ці аспекти можуть бути враховані в майбутніх версіях програмно продукту.

Підводячи підсумок, можна сказати, що розроблений програмний продукт підтвердив можливість застосування теоретико-ігрових методів для моделювання конфліктних ситуацій у легкій для сприйняття формі. Отримані результати можуть бути використані як у навчальному процесі для вивчення теорії ігор, так і в практичних ситуаціях — для аналізу конкурентної взаємодії між суб'єктами ринку. Надалі планується розширення функціональності системи та впровадження адаптивних аналітичних моделей для ще більш гнучкого застосування в реальних умовах.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Савченко, В. С. *Теорія прийняття рішень: навчальний посібник*. Київ: Центр учбової літератури, 2019. 280 с.
2. Кушнір, О. І. *Теорія конфліктів: навчальний посібник*. — Київ: Видавництво «Освіта», 2018. — С. 45-47.
3. Осборн М. *Теорія ігор* / М. Осборн; пер. з англ. О. Іваненко. — Київ : Основи, 2004. — 375 с.
4. Хаскі, Дж. *Конфлікти і теорія ігор: сучасний підхід* / Дж. Хаскі. — Київ : Наукова думка, 2012. — 220 с.
5. Неш Дж. Ф. *Теорія ігор* / Дж. Ф. Неш // *Історія теорії ігор : зб. наук. праць* / за ред. І. П. Прізвище. — Київ : Видавництво, 20XX. — С. 10–30.
6. Борель Е. *Про засади теорії ігор* / Е. Борель // *Мат. збірник* / ред. І. Іванов. — Київ : Наукова думка, 1970. — Т. XX, С. 100–120.
7. Wald A. *Statistical Decision Functions*. — New York : Wiley, 1950. — 212 р.
8. von Neumann J. *Zur Theorie der Gesellschaftsspiele* // *Mathematische Annalen*. — 1928. — Bd. 100. — S. 295–320.
9. von Neumann J., Morgenstern O. *Theory of Games and Economic Behavior*. — Princeton : Princeton University Press, 1944. — 625 p.
10. Nash J. F. *Non-Cooperative Games*. — Ph.D. Dissertation, Princeton University, 1950. — 28 p.
11. Галіяш Н. І., Кліменко А. О. *Теорія ігор: навч. посіб.* — Київ: Центр учбової літератури, 2012. — 288 с.
12. Соляник В. П. *Конфліктологія: навч. посібник*. — Київ: Центр учбової літератури, 2014. — 256 с.
13. Кочубей А. Ю. *Багатокритеріальне прийняття рішень: навч. посібник*. — Київ: КНЕУ, 2010. — С. 15–30.
14. Гусак В. І., Кривіцький С. В. *Прийняття управлінських рішень: методи та моделі*. — Київ: Видавництво, 2015. — С. 95–110.

15. Кравченко С. В. *Управління проектами: навч. посібник*. — Київ: КНЕУ, 2013. — С. 120–130
16. McKelvey, R. D., McLennan, A., & Turocy, T. L. *Gambit: Software tools for game theory*. [Online]. Available: <http://www.gambit-project.org>
17. AnyLogic Company. *AnyLogic Simulation Software*. [Online]. Available: <https://www.anylogic.com>
18. Нечаєв, С. В. *Теорія ігор і прийняття рішень: навч. посібник* / С. В. Нечаєв. — Київ: Центр учбової літератури, 2010. — С. 45–60.
19. Неш, Дж. Ф. *Теорія ігор: вибрані праці* / пер. з англ. — К.: Основи, 1997. — С. 45–60.
20. Козак В. В. *Моделювання і прийняття рішень в економіці: навчальний посібник*. — Київ: Центр учбової літератури, 2015. — С. 45–60.
21. Маслов В. М. *Теорія ігор: навчальний посібник*. — Київ: Видавничий дім «Києво-Могилянська академія», 2010. — С. 120–125.
22. Біблюк О. В. *Теорія ігор та прийняття рішень: навч. посібник*. — Київ: Видавничий дім «Кондор», 2018. — С. 110–115.
23. Bass, L., Clements, P., & Kazman, R. *Software Architecture in Practice*, 3rd Edition, Addison-Wesley, 2012.
24. Python Software Foundation. *Built-in Types — Python 3 Documentation*. <https://docs.python.org/3/library/stdtypes.html>
25. Lucid Software Inc. Lucidchart – онлайн-сервіс для побудови блок-схем та діаграм <https://www.lucidchart.com>