КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА
Факультет комп'ютерних наук та кібернетики
Кафедра моделювання складних систем

Кваліфікаційна робота
на здобуття ступеня бакалавра

за спеціальністю 113 прикладна математика

на тему:

НЕЧІТКІ ОПТИМІЗАЦІЙНІ ЗАДАЧІ НА ОСНОВІ ЕКОЛОГО-ЕКОНОМІЧНОЇ МОДЕЛІ

Виконав студент 4-го курсу
Ярослав ДМИТРІЄВ

Науковий керівник:
доцент, кандидат фіз.-мат. наук
Марина КОРОБОВА

Засвідчую, що в цій роботі немає запозичень з праць інших авторів без відповідних посилань.

Студент _____

Роботу заслухано на засіданні кафедрі моделювання складних систем та рекомендовано до захисту, протокол № 10 від 5 червня 2023 року

Завідувач кафедри МСС _____ д.т.н., доц. Дмитро ЧЕРНІЙ

Київ – 2023

ЗМІСТ

РЕФЕРАТ

Обсяг роботи 56 сторінок, 7 рисунків, 4 таблиці, 5 використаних джерел, 3 додатки.

Об’єктом роботи є еколого-економічна модель Леонтьєва-Форда.

Предметом роботи є нечіткі оптимізаційні задачі, що базуються на моделі Леонтьєва-Форда, та їх реалізація мовою програмування *Python*.

Метою роботи є розробка та аналіз моделі Леонтьєва-Форда з використанням методів нечіткої логіки для оптимізації в еколого-економічних системах. Ця мета досягається через вивчення теоретичних основ нечіткості, розробку алгоритму оптимізації на основі цієї моделі та введення нечіткості в параметри моделі.

Результатом роботи є розроблені алгоритми оптимізації, що використовують нечіткі вхідні параметри на основі моделі Леонтьєва-Форда. Ці алгоритми були реалізовані мовою програмування *Python* і можуть використовуватися для аналізу різних еколого-економічних сценаріїв. Також досягнуто глибше розуміння ролі та значимості нечіткості в економічних моделях.

Отримані результати повністю відповідають темі дипломної роботи.

ВСТУП

В епоху глобалізації та інтенсивної конкуренції у всіх сферах діяльності виробничі підходи, що враховують як економічні, так і екологічні аспекти, стають все більш важливими. Нехтування екологічними параметрами у виробничій діяльності може призвести до негативних наслідків, які впливають не лише на навколишнє середовище, але й на репутацію компанії, а в довгостроковій перспективі, її фінансової стабільності.

Тому вивчення екологічно-економічної моделі Леонтьєва-Форда у нечітких рамках стає особливо актуальним. Ця модель враховує фактор невизначеності, пов'язаний з численним впливом на виробництво, і допомагає компаніям краще зрозуміти, як їх виробничі стратегії впливають на навколишнє середовище.

Більше того, використання цієї моделі може сприяти виробництву екологічно чистих продуктів та технологій, що особливо важливо в контексті збільшення потреб у сталому розвитку. Така стратегія, у свою чергу, може забезпечити компаніям конкурентні переваги на ринку.

Важливо зазначити, що вивчення екологічно-економічної моделі Леонтьєва-Форда може допомогти у вирішенні проблем у реальному світі, пов'язаних з оптимізацією виробничих процесів, зменшенням використання ресурсів та забрудненням навколишнього середовища. Враховуючи складність сучасних виробничих систем та невизначеність багатьох їх параметрів, використання таких математичних моделей абсолютно необхідне для вирішення подібних проблем.

Нарешті, з точки зору наукового внеску, ця робота доповнює та розширює існуючі знання в галузі еколого-економічного моделювання шляхом інтеграції традиційних економічних підходів із сучасними екологічними вимогами. Тому результати цього дослідження не тільки мають

практичне значення, але й сприяють розвитку наукових знань у цій важливій галузі.

АКТУАЛЬНІСТЬ. Важливість вивчення задач нечіткого математичного програмування пояснюється зростанням складності сучасних проблем у різних сферах, таких як наука, промисловість, логістика, фінанси та інші сфери життя. Ці проблеми вимагають використання більш вдосконалених методів аналізу та прийняття рішень, серед яких нечітке математичне програмування є одним із ключових інструментів.

Математичне програмування відіграє вирішальну роль у прийнятті оптимальних рішень шляхом моделювання та оптимізації різних процесів та систем. Розробка нових методів та алгоритмів для математичного програмування є важливим та відповідним завданням, яке сприяє прогресу у цій сфері.

Крім того, розробка обчислювальних технологій, таких як штучний інтелект та машинне навчання, збільшує потенційні можливості застосування нечіткого математичного програмування. Це відкриває нові можливості для вирішення більш складних проблем у реальному світі, і, в свою чергу, аргументує актуальність досліджень у цій галузі.

Таким чином, актуальність даної роботи визначається важливістю математичного програмування, зокрема, нечіткого, як інструменту для аналізу та прийняття рішень у складних завданнях оптимізації, потенціалу цієї галузі в контексті розробки нових технологій та методів, а також широкого діапазону застосувань у різних сферах життя.

МЕТА. Вивчення екологічно-економічної моделі Леонтьєва-Форда в контексті нечіткого моделювання відіграє важливу роль в екологічній економіці. Метою даної роботи є впровадження та вдосконалення сучасних наукових методів та технологій для аналізу та прогнозування взаємодії виробничих процесів та екологічних систем в умовах нечітко заданих параметрів. Це відкриває нові горизонти для наукових знань і має значне практичне значення, допомагаючи покращити управління економічною та

екологічною діяльністю підприємств. Необхідно створити розуміння процесів, що відбуваються в економічній сфері та екологічному середовищі, а також щодо впливу останніх на виробничу діяльність. Ставиться мета дослідження механізмів, що регулюють ці взаємодії, та визначити, як вони можуть бути оптимізовані для досягнення кращих екологічних та економічних результатів.

У роботі акцентується на використанні нечіткого моделювання як інструменту для розробки більш точних та динамічних моделей промислової активності та її впливу на довкілля. Для цього створюються моделі, які можуть передбачити поведінку економічних та екологічних систем, враховуючи реальну невизначеність та нестабільність цих систем. Отже, необхідно не лише зрозуміти сутність подібних процесів, але й надати відповідні інструменти для їх ефективного управління. Це означає розробку нових стратегій, політики та рекомендацій, які допоможуть бізнесу та владі забезпечити стабільний та збалансований розвиток у контексті відповідального екологічного управління.

ОБ’ЄКТ І МЕТОДИ ДОСЛІДЖЕННЯ. Об'єктом дослідження даної роботи є комплексні взаємодії між виробничими процесами та екологічними системами, які можуть бути змодельовані за допомогою екологічно-економічної моделі Леонтьєва-Форда. Необхідно зрозуміти, який взаємовплив чинять економічна діяльність та стан екосистеми, і як ці взаємодії можуть бути оптимізовані для досягнення більш вигідних характеристик, а також сталого розвитку.

Методи, які використовуються, починаються з нечіткого моделювання. Це математичний підхід, який дозволяє моделювати взаємодії в системах, де є значна невизначеність або нестабільність. Наприклад, внаслідок зміни клімату або впливу економічної діяльності на екологію. Нечіткі моделі допомагають краще зрозуміти подібні процеси і прогнозувати їхні наслідки.

Другий метод, який використовується, – це оптимізація за допомогою математичного програмування. Цей інструментарій використовується для

визначення найкращих можливих рішень в рамках заданих економічних і екологічних обмежень. Це дозволяє розробляти стратегії та рекомендації, які можуть допомогти покращити економічну ефективність, зменшити екологічну шкоду і сприяти сталому розвитку.

У розробці використовуються різні засоби, основним з яких є програмне забезпечення для обчислень. У роботі використовується мова програмування Python, яка має потужні бібліотеки для розв'язування задач оптимізації (*scipy*) та візуалізації даних (*matplotlib*). Також використовується різноманітні методики математичного моделювання для створення і аналізу моделей, що дозволяє краще зрозуміти сутність взаємодії між виробництвом і екологічною сферою, а також оптимально впливати на цю взаємодію.

СФЕРИ. Можливі сфери застосування результатів цього дослідження доволі широкі і включають:

1. Промисловість. Запропонована в роботі модель може бути використана для планування та оптимізації виробничих процесів, враховуючи їх вплив на навколишнє середовище. Промислові підприємства можуть використовувати ці результати, щоб зменшити негативний вплив своєї виробничої діяльності на довкілля, при цьому зберігаючи або навіть підвищуючи свою виробничу ефективність.
2. Екологічне планування і управління. Моделі та задачі, які розглядаються в роботі, можуть використовуватися організаціями управління для оцінки впливу різних промислових заходів на екосистему та розробки ефективної політики та правил.
3. Управління природними ресурсами. Моделі можуть бути використані для прогнозування впливу використання природних ресурсів на екосистему та планування використання цих ресурсів таким чином, щоб мінімізувати шкоду навколишньому середовищу.
4. Наукові дослідження. Результати даного дослідження можуть бути використані в інших наукових дослідженнях, які вивчають

взаємозв'язок між промисловістю та екологією. Це може включати розробку нових моделей, методів аналізу або стратегій оптимізації.

5. Освіта. Розуміння взаємозв'язків між економікою та довкіллям є вирішальним компонентом освіти зі сталого розвитку. Дана робота може слугувати фундаментом для курсів та уроків у цій галузі.

РОЗДІЛ 1. ОСНОВНІ ТЕОРЕТИЧНІ ПОНЯТТЯ

1.1 Статична модель Леонтьєва-Форда

Однією з перших міждисциплінарних моделей, які включають взаємопов'язаність економіки та навколишнього середовища, була модель, запропонована В. Леонтьєвим та Д. Фордом. Модель має балансовий характер, узагальнює класичну модель «витрати-випуск» та охоплює дві групи галузей: основне виробництво (виробництво матеріалів) та допоміжне виробництво (види економічної діяльності, що стосуються поводження із шкідливими відходами). Основні умови моделі виражаються через систему рівнянь:

$$\begin{aligned} x_1 &= A_{11}x_1 + A_{12}x_2 + y_1, \\ x_2 &= A_{21}x_1 + A_{22}x_2 - y_2. \end{aligned} \tag{1.1}$$

В цій системі $x_1 = (x_1^1, x_2^1, \dots, x_n^1)^T$ – вектор колонка валового випуску продукції, $y_1 = (y_1^1, y_2^1, \dots, y_n^1)^T$ – вектор колонка кінцевого продукту (споживання), $x_2 = (x_1^2, x_2^2, \dots, x_m^2)^T$ – вектор-колонка обсягів знищених забруднювачів, $y_2 = (y_1^2, y_2^2, \dots, y_m^2)^T$ – вектор-колонка обсягів незнищенних забруднювачів; $A_{11} = (a_{ij}^{11})_1^n$ – квадратна матриця затрат продукції i на одиницю випуску продукції j, $A_{12} = (a_{ig}^{12})_{i,g=1}^{n,m}$ – прямокутна матриця затрат продукції i на одиницю знищення забруднювачів g, $A_{21} = (a_{kj}^{21})_{k,j=1}^{m,n}$ – прямокутна матриця випуску забруднювачів k на одиницю виробництва продукції j, $A_{22} = (a_{kg}^{22})_1^m$ – квадратна матриця випуску забруднювачів k на одиницю знищення забруднювачів g.

Модель (1.1) одержується, коли записати баланси для кожної галузі основного виробництва (матеріальне виробництво):

$$x_i^1 = \sum_{j=1}^{n} x_{ij}^{11} + \sum_{g=1}^{m} x_{ig}^{12} + y_i^1, \qquad i = 1, 2, \dots, n, \tag{1.2}$$

та для кожної галузі допоміжного виробництва (знищення забруднювачів):

$$x_k^2 = \sum_{j=1}^{n} x_{kj}^{21} + \sum_{g=1}^{m} x_{kg}^{22} - y_k^2, \qquad k = 1, 2, \dots, m, \tag{1.3}$$

де x_{ij}^{11} – затрати i-ої продукції в j-й галузі основного виробництва, x_{ig}^{12} – затрати i-ої продукції в g-ій галузі допоміжного виробництва, x_{kj}^{21} – випуск k-го забруднювача j-ою галуззю основного виробництва, x_{kg}^{22} – випуск k-го забруднювача g-ою галуззю допоміжного виробництва.

Співвідношення (1.2) та (1.3) характеризують залежності між $2(n+m)+(n+m)^2$ величинами $x_i^1,\ y_i^1,\ x_k^2,\ y_k^2,\ x_{ij}^{11},\ x_{ig}^{12},\ x_{kj}^{21},\ x_{kg}^{22}$. Вони мають дуже загальний характер. Для побудови математичної моделі головним є припущення про те, що $x_{ij}^{11},\ x_{ig}^{12},\ x_{kj}^{21}$ та x_{kg}^{22} – функції від обсягів виробництва відповідної галузі:

$$\begin{gathered} x_{ij}^{11} = \varphi_{ij}^{11}\left(x_j^1\right), \\ x_{ig}^{12} = \varphi_{ig}^{12}\left(x_g^2\right), \\ x_{kj}^{21} = \varphi_{kj}^{21}\left(x_j^1\right), \\ x_{kg}^{22} = \varphi_{kg}^{22}\left(x_g^2\right), \\ i, j = 1, 2, \dots, n, \\ k, g = 1, 2, \dots, m. \end{gathered} \tag{1.4.}$$

Найпростіша версія моделі Леонтьєва-Форда передбачає пропорційну залежність між виробничими витратами та обсягом виробництва, використовуючи лінійні однорідні функції для виробничих витрат:

$$x_{ij}^{11} = a_{ij}^{11} x_j^1,\ x_{ig}^{12} = a_{ig}^{12} x_g^2,\ x_{kj}^{21} = a_{kj}^{21} x_j^1,\ x_{kg}^{22} = a_{kg}^{22} x_g^2. \tag{1.5}$$

Коефіцієнти пропорційності $a_{ij}^{11} \geq 0,\ a_{ig}^{12} \geq 0,\ a_{kj}^{21} \geq 0$ та $a_{kg}^{22} \geq 0$ називають коефіцієнтами відповідно прямих затрат продукції i на виробництво одиниці продукції j, прямих затрат продукції i при знищенні одиниці забруднювачів g, прямого випуску k-го забруднювача при виробництві одиниці продукції j, прямого випуску k-го забруднювача при знищенні одиниці забруднювача g. Ці коефіцієнти в сукупності і утворюють матриці:

$$A_{11} = (a_{ij}^{11})_1^n,$$

$$A_{12} = (a_{ig}^{12})_{i,g=1}^{n,m},$$

$$A_{21} = (a_{kj}^{21})_{k,j=1}^{m,n},$$

$$A_{22} = (a_{kg}^{22})_{1}^{m}.$$

Підставляючи значення x_{ij}^{11}, x_{ig}^{12}, x_{kj}^{21} та x_{kg}^{22} з (1.5) у (1.2), (1.3), одержуємо систему $n+m$ лінійних алгебраїчних рівнянь з $2(n+m)$ змінними $x_i^1,\ y_i^1,\ x_k^2,\ y_k^2$:

$$x_i^1 = \sum_{j=1}^{n} a_{ij}^{11} x_j^1 + \sum_{g=1}^{m} a_{ig}^{12} x_g^2 + y_i^1, \qquad i = 1, 2, \dots, n, \tag{1.6}$$

$$x_k^2 = \sum_{j=1}^{n} a_{kj}^{21} x_j^1 + \sum_{g=1}^{m} a_{kg}^{22} x_g^2 - y_k^2, \qquad k = 1, 2, \dots, m. \tag{1.7}$$

У векторно-матричній формі система рівнянь (1.6), (1.7) набуває вигляду (1.1).

Припускаючи, що коефіцієнти $a_{ij}^{11} \geq 0,\ a_{ig}^{12} \geq 0,\ a_{kj}^{21} \geq 0,\ a_{kg}^{22} \geq 0$, ми неявно поширюємо на всі види виробничої діяльності (матеріальне виробництво та знищення забруднювачів) припущення (гіпотези) основної моделі міжгалузевого балансу (кількість технологічних способів дорівнює кількості видів продукції, а в кожному технологічному способі виробляється лише один вид продукції). І надалі будемо вважати матриці A_{11}, A_{12}, A_{21} та A_{22} невід'ємними:

$$A_{11} \geq 0,$$

$$A_{12} \geq 0,$$

$$A_{21} \geq 0,$$

$$A_{22} \geq 0.$$

У прикладних моделях номенклатура знищуваних забруднювачів менша номенклатури існуючих забруднювачів. Розширення першої номенклатури відбувається тоді, коли концентрація нових забруднювачів стає істотною з точки зору умов життєдіяльності або виробництва та коли

створюються технічні і економічні можливості для боротьби з наявними забруднювачами.

Також варто зауважити, що економічний зміст моделі Леонтьєва-Форда потребує, аби всі її змінні були невід'ємними, тобто

$$\begin{aligned} x_i^1 &\geq 0, \\ x_k^2 &\geq 0, \\ y_i^1 &\geq 0, \\ y_k^2 &\geq 0. \end{aligned} \tag{1.8}$$

Зауваження. Міжгалузеву модель Леонтьєва-Форда (1.1), (1.8) можна також розглядати як узагальнення класичної схеми міжгалузевого балансу на випадок відкритої економічної системи, коли чистий імпорт i окремих видів продукції перевищує невиробниче споживання y (тобто ця продукція імпортується ще й для виробничого споживання). В цьому випадку кінцева продукція y для цих галузей є від'ємною: $y = y - i \leq 0$. Тоді, групуючи галузі з додатною кінцевою продукцією в блок (x_1, y_1), а інші галузі (з не додатною кінцевою продукцією) – в блок $(x_2, -y_2)$, одержимо матриці прямих матеріальних затрат $A_{11} \geq 0,\ A_{12} \geq 0,\ A_{21} \geq 0,\ A_{22} \geq 0$ і модель міжгалузевого балансу у вигляді (1.1), де $y_1 > 0,\ y_2 \geq 0$. Тому одержані результати з дослідження моделі Леонтьєва-Форда можуть бути використані і для дослідження міжгалузевих балансів відкритих економік.

1.2. Умови існування невід'ємних розв'язків

Взаємодія між людським суспільством та природним середовищем слід розглядати як частину єдиної екологічно-економічної системи на будь-якому рівні. Вважається, що найбільш успішне включення екологічного фактора в економічно-математичних моделях в лінійній економіці було досягнуто саме за допомогою моделі «витрати-випуск» Леонтьєва-Форда (1.1).

Усі матриці системи лінійних рівнянь (1.1) вважаються невід'ємними, а економічний зміст моделі Леонтьєва-Форда вимагає, щоб усі змінні її були невід'ємними.

Розв’яжемо формально систему лінійних алгебраїчних рівнянь (1.1) двома способами. За першим способом спочатку знайдемо x_2 з другого рівняння, підставимо знайдене x_2 в перше рівняння і розв’яжемо його відносно x_1. За другим способом спочатку знайдемо x_1 з першого рівняння, підставимо знайдене x_1 в друге рівняння і розв’яжемо його відносно x_2. В результаті одержимо:

$$x_1 = (E_1 - A_1)^{-1}[y_1 - A_{12}(E_2 - A_{22})^{-1}y_2],$$
$$x_2 = (E_2 - A_2)^{-1}[A_{21}(E_1 - A_{11})^{-1}y_1 - y_2],$$

де E_1 та E_2 – діагональні одиничні матриці відповідно n-го та m-го порядків, A_1 та A_2 – квадратні матриці відповідно n-го та m-го порядків:

$$A_1 = A_{11} + A_{12}(E_2 - A_{22})^{-1}A_{21}, \quad (1.9)$$
$$A_2 = A_{22} + A_{21}(E_1 - A_{11})^{-1}A_{12}. \quad (1.10)$$

Введемо поняття продуктивності моделі. Продуктивній економічній системі відповідала така матриця коефіцієнтів A, яка забезпечувала можливість одержання кінцевої продукції ($y > 0$) при відповідних пропорціях розвитку виробництва ($x \geq 0$). Матриця A називається продуктивною, якщо існує невід’ємний вектор $x \geq 0$, що дає змогу одержати додатний вектор кінцевої продукції, $(E - A)x = y > 0$. Для продуктивності матриці A необхідною та достатньою умовою є невід’ємність матриці $B = (E - A)^{-1} \geq 0$. Якщо при цьому матриця A є нерозкладною, то матриця $B = (E - A)^{-1} > 0$.

Узагальнимо поняття продуктивності матриці на випадок блочної матриці A, яка відповідає системі (1.1):

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \geq 0. \quad (1.11)$$

Будемо вважати невід’ємну блочну матрицю (1.11) *продуктивною*, якщо продуктивними є матриці A_{11}, A_{22}, а також матриці A_1, A_2, які визначаються співвідношеннями (1.9) і (1.10).

Продуктивність матриць A_1 та A_2 означає рентабельність основного та допоміжного виробництва за повним циклом виробництва продукції та за повним циклом знищення забруднювачів.

Якщо матриці A_{11}, A_{22}, A_1 та A_2 продуктивні, то матриці

$$(E_1 - A_{11})^{-1} \geq 0, \quad (E_2 - A_{22})^{-1} \geq 0,$$
$$(E_1 - A_1)^{-1} \geq 0, \quad (E_2 - A_2)^{-1} \geq 0,$$

тобто вони існують і мають невід'ємні елементи.

1.3 Базові поняття теорії нечітких множин

Нечіткі набори – це ключовий інструмент нечіткої логіки, розробленої Лотфі Заде в 1965 році. Вони були створені для моделювання невизначеності та неточності, які часто зустрічаються в реальному світі.

На відміну від традиційного або «чіткого» набору, де елемент або належить до набору, або ні, нечіткий набір дозволяє елементам мати різний ступінь членства. Цей ступінь членства, виміряний від 0 до 1, вказує на те, наскільки «певний» елемент належить до набору. Чим ближче значення до 1, тим більше елемента належить до набору; Значення, близькі до 0, вказують на те, що елемент належить до набору меншою мірою.

Нечіткі множини визначаються через функції належності. Функція приналежності $\mu A(x)$ для нечіткої множини A відображає кожен елемент x в універсальній множині X на значення від 0 до 1. Це можна виразити таким чином:

$$A = \{(x, \mu A(x)) | x \in X\},$$

де x – це елемент універсальної множини X, а $\mu A(x)$ – ступінь належності x до A.

Таким чином, нечіткі набори забезпечують більшу гнучкість у моделюванні ситуацій, які є неточними або невизначеними. Вони широко використовуються в різних сферах, включаючи штучний інтелект, системи прийняття рішень, обробку зображень, медичні системи та багато інших.

Нечіткі числа є окремим випадком нечітких множин, які використовуються для представлення та обробки невизначеності числових значень. Вони представляють «м'який» або неточний діапазон чисел замість одного точного значення.

Подібно до загальних нечітких множин, нечіткі числа визначаються за допомогою функцій належності. Функція приналежності $\mu A(x)$ для нечіткого числа A відображає кожне число x на прямій дійсних чисел у значення від 0 до 1. Це можна виразити таким чином:

$$A = \{(x, \mu A(x)) | x \in R\},$$

де x – це число на реальній числовій лінії R, а $\mu A(x)$ – ступінь належності x до A.

Одним із типів нечітких чисел, які часто використовуються, є трапецієподібні нечіткі числа. Вони визначаються чотирма числами (a, b, c, d), де a і d – ліва і права межі діапазону, а b і c – точки, в яких ступінь приналежності досягає максимуму. Функція належності для трапецієподібного нечіткого числа виглядає так:

$$\begin{cases} \mu A(x) = 0, \text{ якщо } x < a \text{ або } x > d, \\ \mu A(x) = (x - a)/(b - a), \text{ якщо } a \le x < b, \\ \mu A(x) = 1, \text{ якщо } b \le x \le c, \\ \mu A(x) = (d - x)/(d - c), \text{ якщо } c < x \le d. \end{cases} \quad (1.12)$$

Це означає, що ступінь приналежності поступово зростає від a до b, потім залишається постійною від b до c, а потім знову зменшується від c до d.

Також існують інші функції належності, такі як:

- Трикутна функція належності: ця функція має форму трикутника. Вона визначається трьома точками: лівою межею (a), правою межею (c) і вершиною (b), де функція досягає 1.

$$\begin{cases} \mu A(x) = 0, \text{ якщо } x < a \text{ або } x > c, \\ \mu A(x) = (x - a)/(b - a), \text{ якщо } a \le x < b, \\ \mu A(x) = (c - x)/(c - b), \text{ якщо } b \le x < c. \end{cases} \quad (1.13)$$

- Функція належності Гаусса: ця функція має форму класичного (нормального) розподілу Гаусса. Він характеризується двома параметрами: середнім значенням (c) і стандартним відхиленням (σ).

$$\mu A(x) = \exp(-0.5((x - c)/\sigma)^2). \quad (1.14)$$

- Сигмоїдна функція належності: сигмоїдна функція має форму «S» і використовується для представлення процесу, який переходить з одного стану в інший через перехідний період. Вона визначається двома параметрами: точкою переходу (c) і швидкістю переходу (a).

$$\mu A(x) = 1/\big(1 + \exp\big(-a(x - c)\big)\big). \quad (1.15)$$

Незважаючи на те, що специфікація вирішення завдань з нечіткістю може сильно варіюватись у залежності від контексту, загальний алгоритм обробки нечітких даних можна описати таким чином:

1. Визначення нечітких множин: Першим кроком є визначення нечітких наборів, які представляють нечіткість у задачі. Це передбачає вибір функцій належності для представлення ступеня приналежності елементів до цих наборів.
2. Побудова нечіткої моделі: Наступним кроком є побудова моделі, яка описує проблему за допомогою нечітких наборів. Це може передбачати визначення нечітких правил, якщо використовується система нечіткого висновку, або використання нечітких чисел для представлення нечітких значень.
3. Нечіткий висновок: Після побудови моделі нечіткі висновки можна зробити за допомогою таких процесів, як нечітка активація, нечітка агрегація та дефазифікація. Це може включати використання таких методів, як max-min або середнє значення максимуму (mean of maximum, MOM).
4. Інтерпретація результатів: Останнім кроком є інтерпретація отриманих результатів. Це може включати перетворення нечіткого висновку в конкретне числове значення за допомогою

процесу, відомого як дефазифікація, і аналіз цього значення для вирішення початкової проблеми.

Важливо зазначити, що конкретні деталі цих кроків можуть значно відрізнятися залежно від конкретного типу завдання та використовуваного методу нечіткого моделювання.

РОЗДІЛ 2. ОПТИМІЗАЦІЙНА ЗАДАЧА НА ОСНОВІ МОДЕЛІ ЛЕОНТЬЄВА-ФОРДА

2.1 Оптимальне функціонування екологічної економіки

Модель Леонтьєва-Форда (1.1) використовується як компонент у складніших екологічно-економічних моделях. Як приклад, продемонструємо використання цієї моделі для побудови виробничої функції екологічної економіки. Очевидно, що екологічна економіка базується на ринкових відносинах, які охоплюють як економічні, так і екологічні компоненти.

Нехай c_1 – вектор-рядок економічної оцінки кінцевої продукції, c_2 – вектор-рядок економічної оцінки збитків від викидів забруднювачів в навколишнє середовище, b – вектор-стовпчик наявних загальних (первинних) ресурсів, $B_1 \geq 0$ та $B_2 \geq 0$ – технологічні матриці витрат цих ресурсів на відповідні види діяльності. Тоді оптимальне функціонування екологічної економіки, зокрема, можна описати такою задачею лінійного програмування:

$$\begin{gathered} c_1 y_1 - c_2 y_2 \to \max, \\ x_1 = A_{11} x_1 + A_{12} x_2 + y_1, \\ x_2 = A_{21} x_1 + A_{22} x_2 - y_2, \\ B_1 x_1 + B_2 x_2 \leq b, \\ x_1 \geq 0, x_2 \geq 0, \\ y_1 \geq 0, y_2 \geq 0. \end{gathered} \tag{2.1}$$

Вважаючи компоненти вектора b параметрами в задачі (2.1), і розв'язуючи цю параметричну задачу лінійного програмування, знаходимо оптимальні розв'язки:

$$x_1^*(b), \qquad x_2^*(b), \qquad y_1^*(b), \qquad y_2^*(b),$$

на основі яких формується еколого-економічна виробнича функція:

$$F(b) = c_1 y_1^*(b) - c_2 y_2^*(b).$$

Якщо в задачі (2.1) зняти обмеження $y_1 \geq 0$ та $y_2 \geq 0$, то цю задачу після виключення змінних y_1 та y_2 можна спростити до такої:

$$\begin{gathered} (c_1(E_1 - A_{11}) - c_2 A_{21}) x_1 + \big(c_1 A_{12} - c_2(E_2 - A_{22})\big) x_2 \to \max, \\ B_1 x_1 + B_2 x_2 \leq b, \\ x_1 \geq 0, x_2 \geq 0. \end{gathered} \tag{2.2}$$

Задача (2.2) описує оптимальне функціонування гіпотетичної екологічної економіки, що може як виробляти продукцію, так і здійснювати політику щодо чистого довкілля. Виробнича функція такої екологічної економіки запишеться у вигляді:

$$F_0(b) = (c_1(E_1 - A_{11}) - c_2A_{21})\bar{x}_1(b) + \big(c_1A_{12} - c_2(E_2 - A_{22})\big)\bar{x}_2(b),$$

де $\bar{x}_1(b)$ та $\bar{x}_2(b)$ – оптимальні розв'язки задачі (2.2).

Так як в задачі (2.2) допустима область задається параметрично, то у випадку, коли метою є лише знаходження параметрів виробничої функції (цільова функція даної задачі), то раціональним способом досягнення такої мети є перехід в задачі (2.2) до двоїстої:

$$pb \to \min,$$
$$pB_1 \geq c_1(E_1 - A_{11}) - c_2A_{21},$$
$$pB_2 \geq c_1A_{12} - c_2(E_2 - A_{22}),$$
$$p \geq 0,$$

де p – вектор цін на ресурси відповідної розмірності. Аналізуючи цю задачу, приходимо до висновку, що достатньою умовою існування її розв'язку є умови:

$$c_1(E_1 - A_{11}) \geq c_2A_{21} \quad \text{та} \quad c_2A_{12} \geq c_2(E_2 - A_{22}),$$

які пов'язують між собою коефіцієнти c_1 та c_2.

Оскільки у двоїстій задачі допустима область визначається однозначно, то від співвідношення компонент параметричного вектора b буде залежати кут нахилу гіперповерхні рівня цільової функції. Отже, отримаємо оптимальне значення цільової функції як функції параметрів вектора b. А оскільки, як відомо, на оптимальних розв'язках значення цільових функцій прямої та двоїстої задач збігаються, то таким чином і отримуємо оптимальну виробничу функцію $F(b)$ як цільову функцію задачі (2.2).

Зауважимо, що подібний підхід можна за певних обставин застосувати і при розв'язуванні задачі з нечіткими обмеженнями.

2.2 Двовимірний випадок

Розглянемо варіант задачі (2.1) при $n = 1,\ m = 1$. Тобто, виробляється один продукт і при цьому «генерується» один вид забруднення. По суті, це означає, що задача (2.1) має розмірність два.

У Додатку А міститься програма, яка використовує модель Леонтьєва-Форда для вирішення задачі лінійного програмування вигляду (2.1). Для вирішення цієї проблеми ми використовуємо бібліотеку *scipy.optimize*. Далі більш детальне пояснення щодо коду:

Спочатку визначаються параметри моделі: A_{11}, A_{12}, A_{21}, A_{22}, C_1, C_2, B_1, B_2та b.

- A_{11}, A_{12}, A_{21}, A_{22} – це технологічні матриці прямих виробничих витрат і прямих викидів забруднювачів відповідно.
- c_1 та c_2 – це лінійні вектори економічної оцінки кінцевих продуктів та збитків від викиду забруднювачів у навколишнє середовище відповідно.
- B_1, B_2 – технологічні матриці витрат ресурсів.
- b – це вектори доступних загальних ресурсів.

Далі задаються коефіцієнти цільової функції та обмеження:

```
c = [-C1*(E1-A11)+C2*A21, (C2*(E2-A22)-C1*A12)]
```

Це вектор коефіцієнтів для цільової функції, яку, згідно із моделлю (2.1), ми максимізуємо: $(c_1(E_1 - A_{11}) - c_2A_{21})x_1 + \big(c_1A_{12} - c_2(E_2 - A_{22})\big)x_2$. Це перетворюється у форму, придатну для використання у функції *linprog*, яка мінімізує вираз, тому ми приймаємо протилежні значення коефіцієнтів.

```
A_ub = [[B1, B2],
        [-E1+A11, -A12],
        [-A21,-E2+A22]]
b_ub = [b,0,0]
```

Це матриця коефіцієнтів та вектор вільних членів обмежень типу нерівності. Вони відповідають обмеженням в задачі (2.1):

$$(E_1 - A_{11})x_1 - A_{12}x_2 \geq 0,$$

$$A_{21}x_1 - (E_2 - A_{22})x_2 \geq 0,$$

$$B_1x_1 + B_2x_2 \leq b.$$

```
x0_bounds = (0, None)
x1_bounds = (0, None)
```

Це обмеження на знаки змінних. У нашій моделі змінні x_1 і x_2 мають бути невід'ємними, тому ми встановлюємо нижні межі рівними нулю, а верхні межі рівні *None* (що відповідає нескінченності).

```
res = linprog(c, A_ub=A_ub, b_ub=b_ub, bounds=[x0_bounds, x1_bounds], method='highs')
```

Це виклик функції *linprog*, яка розв'язує задачу лінійного програмування. Передаємо коефіцієнти цільової функції, обмеження нерівності та межі змінних знаків. Для методу оптимізації встановлено значення *highs*, що є одним із найновіших і найефективніших солверів для лінійного програмування. Результати виводяться на екран та відображаються на графіку.

При значеннях параметрів моделі $A_{11} = 0.3$, $A_{12} = 0.4$, $A_{21} = 0.5$, $A_{22} = 0.6$, $c_1 = 0.9$, $c_2 = 0.8$, $B_1 = B_2 = 4$, $b = 80$, $E_1 = E_2 = 1$, код дає такі результати:

```
Отпимальне значення: 4.6
X1, X2: [20.  0.]
```

У результатах ми бачимо оптимальну величину цільової функції та вектор змінних, які цю величину забезпечують.

На графіку (рис. 2.1) зображено області, які відповідають обмеженням задачі. Область допустимих рішень знаходиться в перетині цих областей. Червона точка вказує на оптимальне рішення задачі.

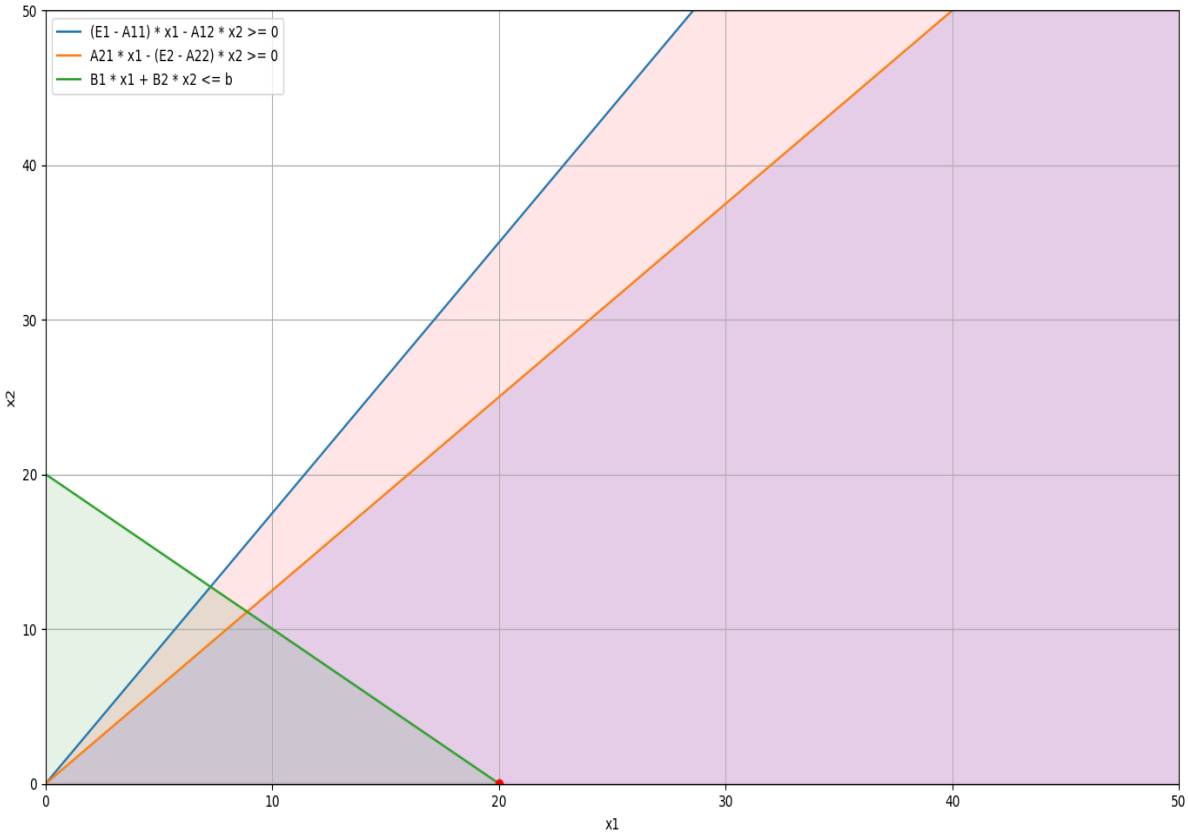

Рисунок 2.1. Допустима область задачі (2.1)

2.3 Тривимірний випадок

Щоб забезпечити більшу гнучкість та точність у нашій моделі та дозволити нам краще пояснити різноманітність факторів, що впливають на екологію та економіку, еколого-економічна виробнича модель Леонтьєва-Форда буде переведена з одновимірних матриць до матриць розмірності $n = 2,\ m = 1$. Фактично, це означає, що задача (2.1) матиме розмірність три.

Використання матриць більшого розміру має ряд переваг у порівнянні з одинарними матрицями. По-перше, вони дозволяють нам більш детально моделювати ситуацію, враховуючи більше взаємозв'язків між різними вхідними та вихідними змінними. Це може включати різні види ресурсів, різні економічні сектори, різні впливи на навколишнє середовище тощо.

По-друге, використовуючи більші розмірності матриць, ми можемо уникнути деяких обмежень. Наприклад, у матрицях меншої розмірності

можна зіткнутися з проблемами масштабування, де більші значення одного фактора можуть затьмарювати вплив менших факторів. У випадку збільшення розмірності задачі можна точніше моделювати ці взаємозв'язки.

По-третє, матриці більшого розміру можуть допомогти більш точно моделювати реальний світ. У реальних сценаріях часто доводиться мати справу з великою кількістю змінних, і використання моделей відповідних розмірностей допомагає краще врахувати цю складність.

Таким чином, розширення моделі для використання матриць з розмірами $n = 2,\ m = 1$ – важливий крок до більш реалістичної та точної моделі. Це допоможе нам краще зрозуміти відносини між економікою та екологією, а також допоможе нам розробити більш ефективні стратегії збалансованого розвитку.

Тепер параметри задачі (2.1) є такими:

$$A_{11} = \begin{pmatrix} a_{11}^{11} & a_{12}^{11} \\ a_{21}^{11} & a_{22}^{11} \end{pmatrix} \text{ – матриця;}$$

$$A_{12} = \begin{pmatrix} a_1^{12} \\ a_2^{12} \end{pmatrix} \text{ – вектор;}$$

$$A_{21} = (a_1^{21} \quad a_2^{21}) \text{ – вектор;}$$

$$A_{22} = a_{22} \text{ – скаляр;}$$

$$B_1 = (b_1^1 \quad b_2^1) \text{ – вектор;}$$

$$B_2 = b_2 \text{ – скаляр;}$$

$$E_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \text{ – матриця;}$$

$$E_2 = 1;$$

b –скаляр;

$$x_1 = \begin{pmatrix} x_{11} \\ x_{12} \end{pmatrix},\ y_1 = \begin{pmatrix} y_{11} \\ y_{12} \end{pmatrix} \text{ – вектори;}$$

$$c_1 = (c_{11} \quad c_{12}) \text{ – вектор;}$$

$x_2,\ y_2,\ c_2$ – скаляри.

Підставивши такі змінні у задачу (2.1), отримаємо:

$$(c_{11} \quad c_{12}) \begin{pmatrix} y_{11} \\ y_{12} \end{pmatrix} - c_2 y_2 \to \max,$$

$$\begin{pmatrix} x_{11} \\ x_{12} \end{pmatrix} = \begin{pmatrix} a_{11}^{11} & a_{12}^{11} \\ a_{21}^{11} & a_{22}^{11} \end{pmatrix} \begin{pmatrix} x_{11} \\ x_{12} \end{pmatrix} + \begin{pmatrix} a_1^{12} \\ a_2^{12} \end{pmatrix} x_2 + \begin{pmatrix} y_{11} \\ y_{12} \end{pmatrix},$$

$$x_2 = (a_1^{21} \quad a_2^{21}) \begin{pmatrix} x_{11} \\ x_{12} \end{pmatrix} + a_{22} x_2 - y_2,$$

$$(b_1^1 \quad b_2^1) \begin{pmatrix} x_{11} \\ x_{12} \end{pmatrix} + b_2 x_2 \le b,$$

$$\begin{pmatrix} x_{11} \\ x_{12} \end{pmatrix} \ge 0, \qquad x_2 \ge 0,$$

$$\begin{pmatrix} y_{11} \\ y_{12} \end{pmatrix} \ge 0, \qquad y_2 \ge 0.$$

Наступним кроком розпишемо модель покомпонентно. Перемножимо вектори c_1 і y_1 запишемо цільову функцію задачі:

$$c_{11} y_{11} + c_{12} y_{12} - c_2 y_2 \to \max.$$

Далі потрібно виразити y_{11}, y_{12} і y_2 з відповідних обмежень. Спочатку виразимо y_{11} та y_{12}:

$$\begin{pmatrix} x_{11} \\ x_{12} \end{pmatrix} = \begin{pmatrix} a_{11}^{11} & a_{12}^{11} \\ a_{21}^{11} & a_{22}^{11} \end{pmatrix} \begin{pmatrix} x_{11} \\ x_{12} \end{pmatrix} + \begin{pmatrix} a_1^{12} \\ a_2^{12} \end{pmatrix} x_2 + \begin{pmatrix} y_{11} \\ y_{12} \end{pmatrix};$$

$$\begin{pmatrix} x_{11} \\ x_{12} \end{pmatrix} - \begin{pmatrix} a_{11}^{11} & a_{12}^{11} \\ a_{21}^{11} & a_{22}^{11} \end{pmatrix} \begin{pmatrix} x_{11} \\ x_{12} \end{pmatrix} - \begin{pmatrix} a_1^{12} \\ a_2^{12} \end{pmatrix} x_2 = \begin{pmatrix} y_{11} \\ y_{12} \end{pmatrix};$$

$$\left(\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \begin{pmatrix} a_{11}^{11} & a_{12}^{11} \\ a_{21}^{11} & a_{22}^{11} \end{pmatrix} \right) \begin{pmatrix} x_{11} \\ x_{12} \end{pmatrix} - \begin{pmatrix} a_1^{12} \\ a_2^{12} \end{pmatrix} x_2 = \begin{pmatrix} y_{11} \\ y_{12} \end{pmatrix};$$

$$\begin{pmatrix} 1 - a_{11}^{11} & -a_{12}^{11} \\ -a_{21}^{11} & 1 - a_{22}^{11} \end{pmatrix} \begin{pmatrix} x_{11} \\ x_{12} \end{pmatrix} - \begin{pmatrix} a_1^{12} \\ a_2^{12} \end{pmatrix} x_2 = \begin{pmatrix} y_{11} \\ y_{12} \end{pmatrix};$$

$$(1 - a_{11}^{11}) x_{11} - a_{12}^{11} x_{12} - a_1^{12} x_2 = y_{11};$$

$$-a_{21}^{11} x_{11} + (1 - a_{22}^{11}) x_{12} - a_2^{12} x_2 = y_{12}.$$

Тепер виразимо y_2:

$$x_2 = (a_1^{21} \quad a_2^{21}) \begin{pmatrix} x_{11} \\ x_{12} \end{pmatrix} + a_{22} x_2 - y_2;$$

$$(a_1^{21} \quad a_2^{21}) \begin{pmatrix} x_{11} \\ x_{12} \end{pmatrix} - x_2 - a_{22} x_2 = y_2;$$

$$a_1^{21} x_{11} + a_2^{21} x_{12} + (a_{22} - 1) x_2 = y_2.$$

Перемножимо вектори B_1 та x_1 щоб отримати останнє обмеження задачі у потрібному вигляді:

$$b_1^1 x_{11} + b_2^1 x_{12} + b_2 x_2 \le b.$$

Наступним кроком підставимо y_{11}, y_{12} та y_2 у цільову функцію та виділимо в ній коефіцієнти при змінних x_{11}, x_{12} та x_2:

$$c_{11}\big((1 - a_{11}^{11})x_{11} - a_{12}^{11}x_{12} - a_1^{12}x_2\big) + c_{12}(-a_{21}^{11}x_{11} + (1 - a_{22}^{11})x_{12} - a_2^{12}x_2) -$$
$$-c_2(a_1^{21}x_{11} + a_2^{21}x_{12} + (a_{22} - 1)x_2) \to \max;$$

$$x_{11}(c_{11}(1 - a_{11}^{11}) - c_{12}a_{21}^{11} - c_2a_1^{21}) + x_{12}(-c_{11}a_{12}^{11} + c_{12}(1 - a_{22}^{11}) - c_2a_2^{21}) +$$
$$+x_2\big(-c_{11}a_1^{12} - c_{12}a_2^{12} - c_2(a_{22} - 1)\big) \to \max.$$

Виконавши всі попередні кроки отримаємо таку модель:

$$(c_{11}(1 - a_{11}^{11}) - c_{12}a_{21}^{11} - c_2a_1^{21})x_{11} + (-c_{11}a_{12}^{11} + c_{12}(1 - a_{22}^{11}) - c_2a_2^{21})x_{12} +$$
$$+\big(-c_{11}a_1^{12} - c_{12}a_2^{12} - c_2(a_{22} - 1)\big)x_2 \to \max.$$

$$\begin{aligned}
(1 - a_{11}^{11})x_{11} - a_{12}^{11}x_{12} - a_1^{12}x_2 &\ge 0,\\
-a_{21}^{11}x_{11} + (1 - a_{22}^{11})x_{12} - a_2^{12}x_2 &\ge 0,\\
a_1^{21}x_{11} + a_2^{21}x_{12} + (a_{22} - 1)x_2 &\ge 0,\\
b_1^1 x_{11} + b_2^1 x_{12} + b_2 x_2 &\le b,\\
x_{11} \ge 0, \qquad x_{12} \ge 0, \qquad x_2 &\ge 0.
\end{aligned} \tag{2.3}$$

Після деталізації моделі у вигляді (2.3) впровадимо її в коді. Перший крок у цьому процесі – конвертація функції максимізації на функцію мінімізації. Це необхідний крок, оскільки *linprog* – це оптимізаційний розв'язувач, доступний в бібліотеці *SciPy Python*, який виконує тільки задачі мінімізації. Зауважимо, що максимізація функції є еквівалентною мінімізації її протилежного значення. Таким чином, ми можемо легко адаптувати задачу для використання з *linprog*, просто змінивши знак цільової функції:

$$\big(c_{12}a_{21}^{11} + c_2a_1^{21} - c_{11}(1 - a_{11}^{11})\big)x_{11} + (c_{11}a_{12}^{11} - c_{12}(1 - a_{22}^{11}) + c_2a_2^{21})x_{12} +$$
$$+\big(c_{11}a_1^{12} + c_{12}a_2^{12} + c_2(a_{22} - 1)\big)x_2 \to \min. \tag{2.4}$$

В коді це реалізовано таким чином:

```
c_coeff = [-(C11 * (1 - A11[0][0]) - C12 * A11[1][0] - C2*A21[0][0]),
           -(-C11*A11[0][1]+C12*(1-A11[1][1])-C2*A21[0][1]),
           -(-C11*A12[0]-C12*A12[1]-C2*(A22-1))]
```

Далі, потрібно вписати в код обмеження для моделі. Щоб успішно реалізувати обмеження у коді, потрібно адаптувати їх до формату, прийнятного для *linprog*. Вона вимагає, щоб усі обмеження були відформатовані у форму $Ax + b \leq 0$. Маємо:

$$-(1 - a_{11}^{11})x_{11} + a_{12}^{11}x_{12} + a_1^{12}x_2 \leq 0,$$

$$a_{21}^{11}x_{11} - (1 - a_{22}^{11})x_{12} + a_2^{12}x_2 \leq 0,$$

$$-a_1^{21}x_{11} - a_2^{21}x_{12} - (a_{22} - 1)x_2 \leq 0,$$

$$b_1^1 x_{11} + b_2^1 x_{12} + b_2 x_2 \leq b,$$

$$x_{11} \geq 0, \qquad x_{12} \geq 0, \qquad x_2 \geq 0.$$

В коді обмеження реалізовані так:

```
A_ub = [[B11[0], B12[0], B2[0]],
        [-1 + A11[0][0], A11[0][1],  A12[0]],
        [A11[1][0], -1 + A11[1][1], A12[1]],
        [-A21[0][0], -A21[0][1], -A22+1]]
b_vector = [b2[0], 0, 0, 0]
x_bounds = [(0, None), (0, None), (0, None)]
```

Використовуючи такі значення змінних:

$$A_{11} = \begin{pmatrix} 0.5 & 0.3 \\ 0.4 & 0.3 \end{pmatrix},$$

$$A_{12} = \begin{pmatrix} 0.3 \\ 0.2 \end{pmatrix},$$

$$A_{21} = (0.3 \quad 0.2),$$

$$A_{22} = a_{22} = 0.2,$$

$$B_1 = (0.6 \quad 0.5),$$

$$B_2 = b_2 = 0.4,$$

$$b = 150,$$

$$c_1 = (0.5 \quad 0.6),$$

$$c_2 = 0.55,$$

код видає такі результати:

```
Optimal Solution= 12.919708029197084 x_values= [109.48905109 113.13868613  69.34306569]
```

Тут «*Optimal Solution*» – це оптимальне значення цільової функції, отримане в результаті розв'язання задачі оптимізації, «*x_values*» – це вектор оптимальних значень змінних x_{11}, x_{12} та x_2, що отримані в результаті розв'язання задачі оптимізації.

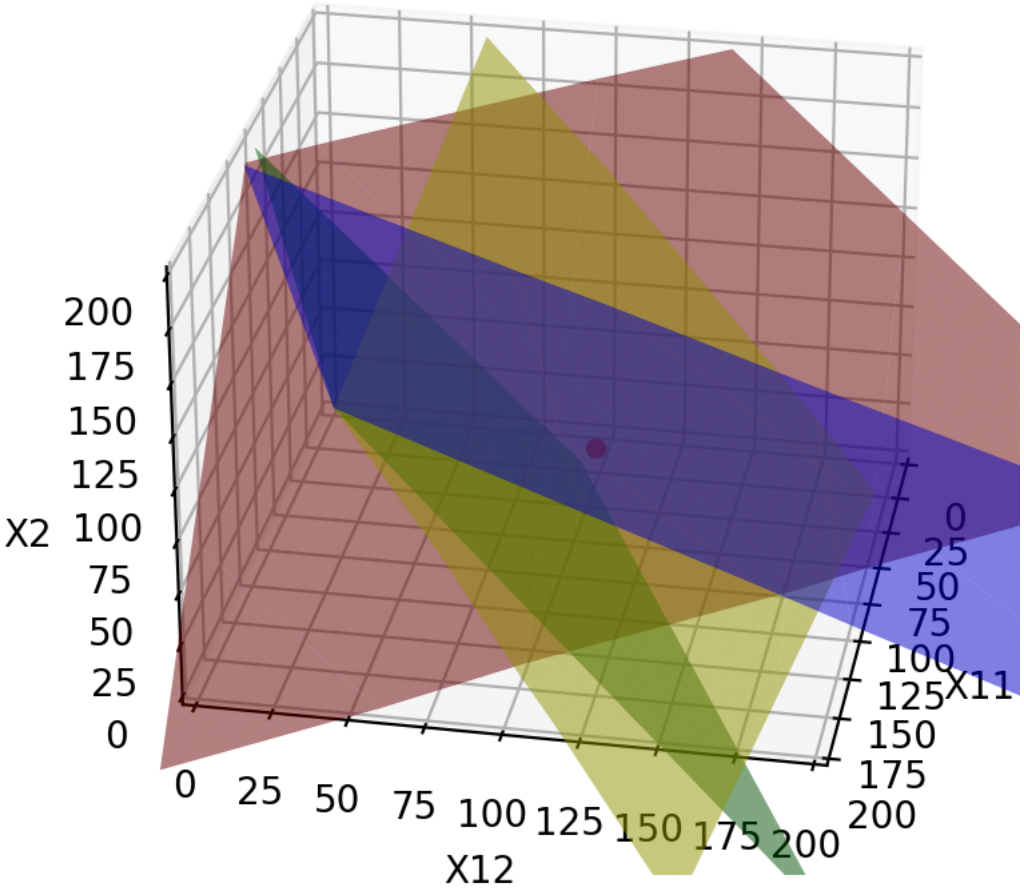

Рисунок 2.2. Оптимальний розв'язок задачі (2.3)

На рис. 2.2 подано графік, на якому відображені оптимальне значення змінних x_{11}, x_{12} і x_2 у вигляді червоної точки та наступні обмеження:

$$(1 - a_{11}^{11})x_{11} - a_{12}^{11}x_{12} - a_1^{12}x_2 \geq 0 \quad \textit{(відображено синім кольором)},$$

$$-a_{21}^{11}x_{11} + (1 - a_{22}^{11})x_{12} - a_2^{12}x_2 \geq 0 \quad \textit{(відображено жовтим кольором)},$$

$$a_1^{21}x_{11} + a_2^{21}x_{12} + (a_{22} - 1)x_2 \geq 0 \quad \textit{(відображено зеленим кольором)},$$

$$b_1^1 x_{11} + b_2^1 x_{12} + b_2 x_2 \leq b \quad \textit{(відображено червоним кольором)}.$$

З повною версією коду можна ознайомитись у Додатку Б.

РОЗДІЛ 3. НЕЧІТКІ ПОСТАНОВКИ ЗАДАЧ НА ОСНОВІ МОДЕЛІ ЛЕОНТЬЄВА-ФОРДА

3.1 Оптимізаційна задача, в якій параметри цільової функції задані нечітко

Після розгляду теоретичних основ нечіткої логіки, включаючи її основні поняття, такі як нечіткі множини, функції приналежності та різні методи представлення та обробки нечіткості, перейдемо до застосування цих знань на практиці. Введемо елемент невизначеності, додавши нечіткість до деяких параметрів нашої задачі. Це дозволить провести дослідження, як нечіткість може вплинути на процес прийняття рішень, і вибрати найкраще можливе рішення, враховуючи цю невизначеність.

З цією метою буде використано бібліотеку мови *Python – skfuzzy*, яка спеціально розроблена для роботи з нечіткими системами. Вона пропонує широкий спектр функцій для роботи з нечіткими множинами, включаючи функції для визначення та візуалізації функцій належності.

Будемо розглядати нечіткість в параметрах c_1 і c_2, які впливають на формування цільової функції оптимізаційної моделі, яка розглядалася в попередньому розділі. Мета – зрозуміти вплив нечіткості на оптимальні рішення проблеми. Це дозволить не тільки краще зрозуміти взаємозв’язок між нечіткістю та процесами прийняття рішень, але й оцінити практичне застосування методів нечіткої логіки в області оптимізації та лінійного програмування.

Розв’язання оптимізаційної задачі з нечіткими параметрами відрізняється від традиційного розв’язання. Введення нечіткості дозволяє більш гнучко реагувати на зміну умов задачі. Врахування нечіткості параметрів дозволяє розглядати множину можливих сценаріїв, що сприяє пошуку більш стійкого та адаптивного розв’язку, замість конкретного оптимального пункту. Це може бути особливо корисним в умовах невизначеності або умов, які швидко змінюються.

Алгоритм розв'язання оптимізаційної задачі з нечіткістю в параметрах відрізняється тим, що потрібно опрацьовувати всі комбінації значень c_1 та c_2, обчислюючи оптимальні рішення для кожної, а також «ймовірність» кожної пари комбінацій параметрів c_1 та c_2. Ця «ймовірність» відображає ступінь придатності комбінації параметрів і визначається як мінімум з двох значень належності до відповідних нечітких наборів для c_1 і c_2. Формула для розрахунку ступеня приналежності виглядає так:

$$\varphi = \min(\mu c_1;\ \mu c_2), \tag{3.1}$$

де μc_1 та μc_2 значення належності до відповідних нечітких наборів.

3.2 Нечіткі параметри цільової функції з трапецієподібними функціями належності

У підрозділі 2.3 було описано функціональний код для розв'язання тривимірної задачі оптимізації, що базується на моделі Леонтьєва-Форда (1.1). Однак, враховуючи, що параметри цільової функції c_1 і c_2 часто виявляються неточними в реальному світі, включення нечіткості у ці параметри, може значно підвищити точність відповідних прогнозів. Як було зазначено в підрозділі 3.1, алгоритм рішення оптимізаційної задачі зазнає деяких змін після введення нечіткості в параметри цільової функції.

У першу чергу впровадимо трапецієподібну нечіткість (1.12) в параметри і c_2. Оскільки у моделі (2.3) c_1 є двовимірним вектором, то й нечіткість буде введена в кожний його компонент, а саме в c_{11} і c_{12}.

Fuzz.trapmf() – це функція бібліотеки *scikit-fuzzy* (підмодуль *fuzz*), яка використовується для створення нечіткої трапецієвидної функції належності. Кожна множина задається чотирма точками, що визначають форму трапеції. Оберемо такі значення: для c_{11}: $[0.3, 0.4, 0.6, 0.8]$, для c_{12}: $[0.35, 0.45, 0.7, 0.95]$ і для c_2: $[0.4, 0.5, 0.6, 0.8]$. Значення трьох параметрів варіюються від 0.1 до 0.8 з кроком 0.05. В коді це реалізовано так:

```
C11_domain = np.arange(0.1, 0.8, 0.05)
C12_domain = np.arange(0.1, 0.8, 0.05)
C2_domain = np.arange(0.1, 0.8, 0.05)

C11_values = fuzz.trapmf(C11_domain, [0.3, 0.4, 0.6, 0.85])
C12_values = fuzz.trapmf(C12_domain, [0.35, 0.45, 0.7, 0.95])
C2_values = fuzz.trapmf(C2_domain, [0.4, 0.5, 0.6, 0.8])
```

На рис. 3.1 показано, як впливає нечіткість на параметри. Можна побачити розподіл значень для кожного параметра з урахуванням введеної нечіткості.

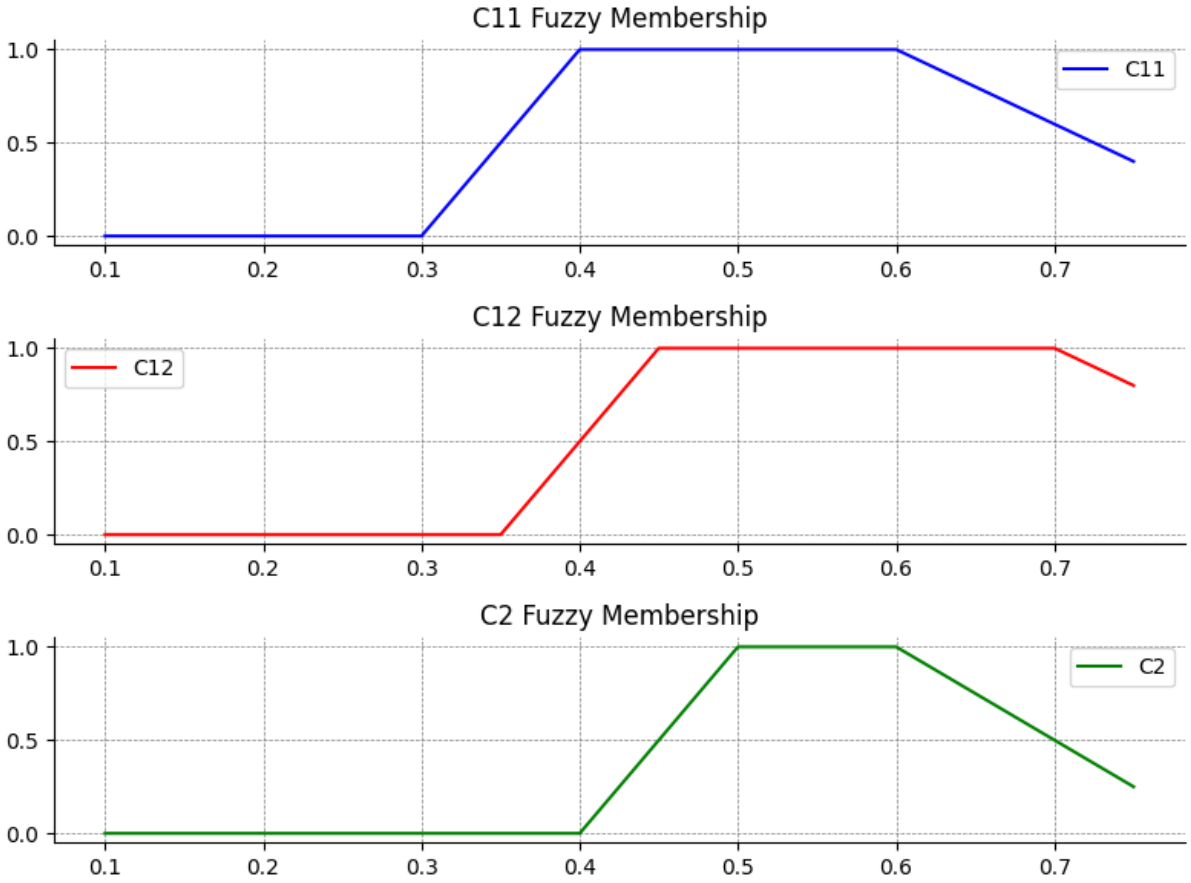

Рисунок 3.1. Трапецієподібні функції належності коефіцієнтів цільової функції

Після введення нечіткого задання параметрів цільової функції задачі (2.3), наступним важливим кроком є генерація всіх можливих комбінацій значень для цих параметрів. Цей процес важливий, оскільки він дає змогу

моделювати широкий спектр можливих сценаріїв виробничого процесу. Цей процес реалізований у такому фрагменті коду:

```
for i in range(len(C11_domain)):
    for j in range(len(C12_domain)):
        for k in range(len(C2_domain)):
```

Під час ітеративного процесу код також обчислює ступінь належності комбінації параметрів, який визначається згідно з формулою (3.1).

Формула для розрахунку ступеня належності в коді представлена так:

```
min(C11_values[i], C12_values[j], C2_values[k])
```

Також, оскільки, ми маємо перебирати всі можливі комбінації параметрів c_{11}, c_{12} і c_2, змін зазнала і цільова функція (2.4). Тепер вона виглядатиме так:

```
c_coeff = [-(C11_domain[i] * (1 - A11[0][0]) - C12_domain[j] * A11[1][0] - C2_domain[k]*A21[0][0]),
          -(-C11_domain[i]*A11[0][1]+C12_domain[j]*(1-A11[1][1])-C2_domain[k]*A21[0][1]),
          -(-C11_domain[i]*A12[0]-C12_domain[j]*A12[1]-C2_domain[k]*(A22-1))]
```

Після ітерації всіх можливих комбінацій параметрів c_{11}, c_{12} і c_2, а також вирішення проблем оптимізації, код генерує таблицю, яка представляє результати цих обчислень.

- c_{11}, c_{12}, c_2 – значення параметрів c_{11}, c_{12} та c_2 для кожної окремої задачі оптимізації. Ці дані відображають конкретний сценарій, який було розглянуто.
- «*Optimal Solution*» – це оптимальне значення цільової функції, отримане в результаті розв'язання задачі оптимізації.
- «*Chance*» – це мінімальне значення ступеня приналежності з трьох параметрів c_{11}, c_{12} та c_2 (відповідно до (3.1)). Цей показник може бути використаний для оцінки ризику, пов'язаної з кожною окремою задачею оптимізації.
- «*x_values*» – це вектор оптимальних значень змінних x_{11}, x_{12} та x_2, що отримані в результаті розв'язання задачі оптимізації.

Додавання значень до таблиці виконується у такому фрагменті коду:

```
y11=(1-A11[0][0])*res.x[0]-A11[0][1]*res.x[1]-A12[0]*res.x[2]
y12=-A11[1][0]*res.x[0]+(1-A11[1][1])*res.x[1]-A12[1]*res.x[2]
y2=A21[0][0]*res.x[0]+A21[0][1]*res.x[1]+(A22-1)*res.x[2]
b=B11[0]*res.x[0]+B12[0]*res.x[1]+B2[0]*res.x[2]

if res.success and y11>=0 and y12>=0 and y2>=0 and b>=b2[0]:
    y22=A21[0][0]*res.x[0]+A21[0][1]*res.x[1]+(A22-1)*res.x[2]
    data_list.append({"C11": C11,
                      "C12": C12,
                      "C2": C2,
                      "Optimal Solution": -res.fun,
                      "Chance": "{:.2f}".format(min(C11_values[i], C12_values[j], C2_values[k])),
                      "x_values": res.x})
```

Важливо зазначити, що *Python* є високорівневою, інтерпретованою мовою програмування, і як така, вона спрямована на легкість використання та читабельність коду, а не на низькорівневий контроль над точністю обчислень та управління пам'яттю. Через це можуть виникнути певні нюанси при роботі з обмеженнями в оптимізаційних задачах.

Більшість оптимізаторів, включаючи ті, що використовуються в бібліотеках *Python*, є ітеративними та використовують певні критерії зупинки для визначення, коли вони повинні припинити пошук оптимального рішення. Ці критерії зупинки можуть викликати помилки, якщо вони дуже жорсткі. Саме тому перед додаванням даних до таблиці код ще раз перевіряє, чи оптимізатор дійсно знайшов рішення, що відповідає всім обмеженням.

Результати обчислень по оптимізаційній задачі з трапецієподібними функціями належності для коефіцієнтів цільової функції наведено в таблиці 3.1.

У таблиці 3.1 продемонстровано тільки 30 комбінацій c_{11}, c_{12} та c_2, відсортованих за максимальним значенням стовпця *«Chance»*. Всього код видав оптимальні рішення для 2669 комбінацій c_{11}, c_{12} та c_2.

Таблиця 3.1

	C11	C12	C2	Optimal Solut	Chance	x_values
1	0.6	0.7	0.55	19.71	1.00	[104.65116279 174.41860465 0.]
2	0.5	0.6	0.55	12.92	1.00	[109.48905109 113.13868613 69.34306569]
3	0.4	0.6	0.5	15.00	1.00	[104.65116279 174.41860465 0.]
4	0.4	0.45	0.6	9.69	1.00	[109.48905109 113.13868613 69.34306569]
5	0.4	0.45	0.55	9.69	1.00	[109.48905109 113.13868613 69.34306569]
6	0.4	0.45	0.5	9.69	1.00	[109.48905109 113.13868613 69.34306569]
7	0.4	0.6	0.55	12.92	1.00	[109.48905109 113.13868613 69.34306569]
8	0.55	0.45	0.6	9.69	1.00	[109.48905109 113.13868613 69.34306569]
9	0.6	0.65	0.5	19.01	1.00	[104.65116279 174.41860465 0.]
10	0.6	0.65	0.55	15.70	1.00	[104.65116279 174.41860465 0.]
11	0.6	0.65	0.6	14.00	1.00	[109.48905109 113.13868613 69.34306569]
12	0.4	0.6	0.6	12.92	1.00	[109.48905109 113.13868613 69.34306569]
13	0.45	0.7	0.5	23.02	1.00	[104.65116279 174.41860465 0.]
14	0.55	0.5	0.5	10.77	1.00	[109.48905109 113.13868613 69.34306569]
15	0.55	0.5	0.55	10.77	1.00	[109.48905109 113.13868613 69.34306569]
16	0.45	0.7	0.55	19.71	1.00	[104.65116279 174.41860465 0.]
17	0.6	0.7	0.5	23.02	1.00	[104.65116279 174.41860465 0.]
18	0.6	0.45	0.55	9.69	1.00	[109.48905109 113.13868613 69.34306569]
19	0.55	0.5	0.6	10.77	1.00	[109.48905109 113.13868613 69.34306569]
20	0.6	0.5	0.55	10.77	1.00	[109.48905109 113.13868613 69.34306569]
21	0.6	0.5	0.5	10.77	1.00	[109.48905109 113.13868613 69.34306569]
22	0.5	0.7	0.55	19.71	1.00	[104.65116279 174.41860465 0.]
23	0.5	0.65	0.6	14.00	1.00	[109.48905109 113.13868613 69.34306569]
24	0.5	0.65	0.5	19.01	1.00	[104.65116279 174.41860465 0.]
25	0.4	0.7	0.55	19.71	1.00	[104.65116279 174.41860465 0.]
26	0.6	0.6	0.6	12.92	1.00	[109.48905109 113.13868613 69.34306569]
27	0.45	0.45	0.5	9.69	1.00	[109.48905109 113.13868613 69.34306569]
28	0.45	0.45	0.55	9.69	1.00	[109.48905109 113.13868613 69.34306569]
29	0.6	0.55	0.6	11.84	1.00	[109.48905109 113.13868613 69.34306569]
30	0.45	0.6	0.6	12.92	1.00	[109.48905109 113.13868613 69.34306569]

Після таблиці код видає графік (рис. 3.2), на якому відображені оптимальні значення змінних x_{11}, x_{12} і x_2 у вигляді червоних точок, а також обмеження:

$$(1 - a_{11}^{11})x_{11} - a_{12}^{11}x_{12} - a_1^{12}x_2 \geq 0 \quad \textit{(відображено синім кольором)},$$

$$-a_{21}^{11}x_{11} + (1 - a_{22}^{11})x_{12} - a_2^{12}x_2 \geq 0 \quad \textit{(відображено жовтим кольором)},$$

$$a_1^{21}x_{11} + a_2^{21}x_{12} + (a_{22} - 1)x_2 \geq 0 \quad \textit{(відображено зеленим кольором)},$$

$$b_1^1 x_{11} + b_2^1 x_{12} + b_2 x_2 \leq b \quad \textit{(відображено червоним кольором)}.$$

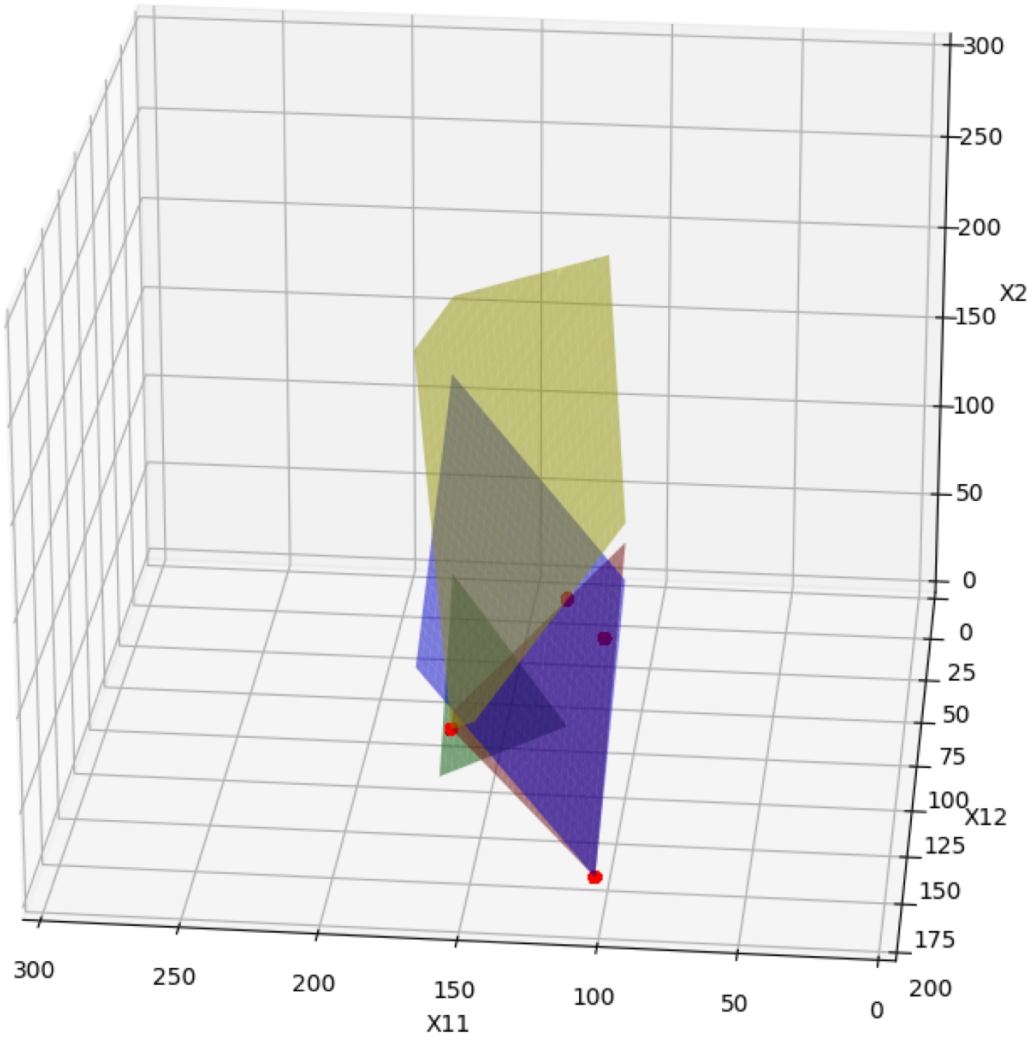

Рисунок 3.2. Оптимальний розв’язок задачі (2.3) у випадку трапецієподібних функцій належності коефіцієнтів цільової функції

З повним варіантом коду можна ознайомитись у Додатку В.

3.3 Нечіткі параметри цільової функції з трикутними функціями належності

Трикутна нечіткість (1.13) може бути корисною для задач оптимізації, особливо при роботі з невизначеністю та варіативністю параметрів. Вона пропонує простоту і інтуїтивність, які випливають з її лінійної форми зростання або спадання, що робить її зручною для інтерпретації та використання в практичних застосуваннях.

Оскільки трикутні нечіткі числа можуть представляти різний ступінь мінливості вхідних даних, вони відображають більшу випадковість порівняно з трапецієподібними числами, у яких значення функції належності залишаються постійними протягом певного інтервалу.

Крім того, використання трикутних нечітких чисел у математиці може спростити обчислення порівняно з іншими типами нечітких чисел.

Таким чином, при моделюванні різних ситуацій в задачах оптимізації, які вимагають врахування неоднозначності та мінливості параметрів, трикутна нечіткість може запропонувати цінні переваги.

Модифікуємо код з Додатку В, ввівши трикутну функцію належності в параметри c_{11}, c_{12} та c_2. *Fuzz.trimf()* – це функція бібліотеки *scikit-fuzzy*, яка використовується для створення нечіткої трикутної функції належності. Кожна множина задається трьома точками, що визначають форму трикутника. Оберемо такі значення: для c_{11}: $[0.3, 0.5, 0.7]$, для c_{12}: $[0.35, 0.55, 0.75]$ і для c_2: $[0.4, 0.6, 0.8]$. Значення трьох параметрів варіюють від 0.1 до 0.8 з кроком 0.05. В коді це реалізовано так:

```
C11_domain = np.arange(0.1, 0.8, 0.05)
C12_domain = np.arange(0.1, 0.8, 0.05)
C2_domain = np.arange(0.1, 0.8, 0.05)

C11_values = fuzz.trimf(C11_domain, [0.3, 0.5, 0.7])
C12_values = fuzz.trimf(C12_domain, [0.35, 0.55, 0.75])
C2_values = fuzz.trimf(C2_domain, [0.4, 0.6, 0.8])
```

На рис. 3.3 продемонстровано, як впливає нечіткість на параметри коефіцієнтів цільової функції. Відображається розподіл значень кожного параметра з урахуванням введеної нечіткості.

Після ітерації всіх можливих комбінацій параметрів c_{11}, c_{12} та c_2, а також після розв'язання оптимізаційної проблеми, код генерує таблицю (табл. 3.2).

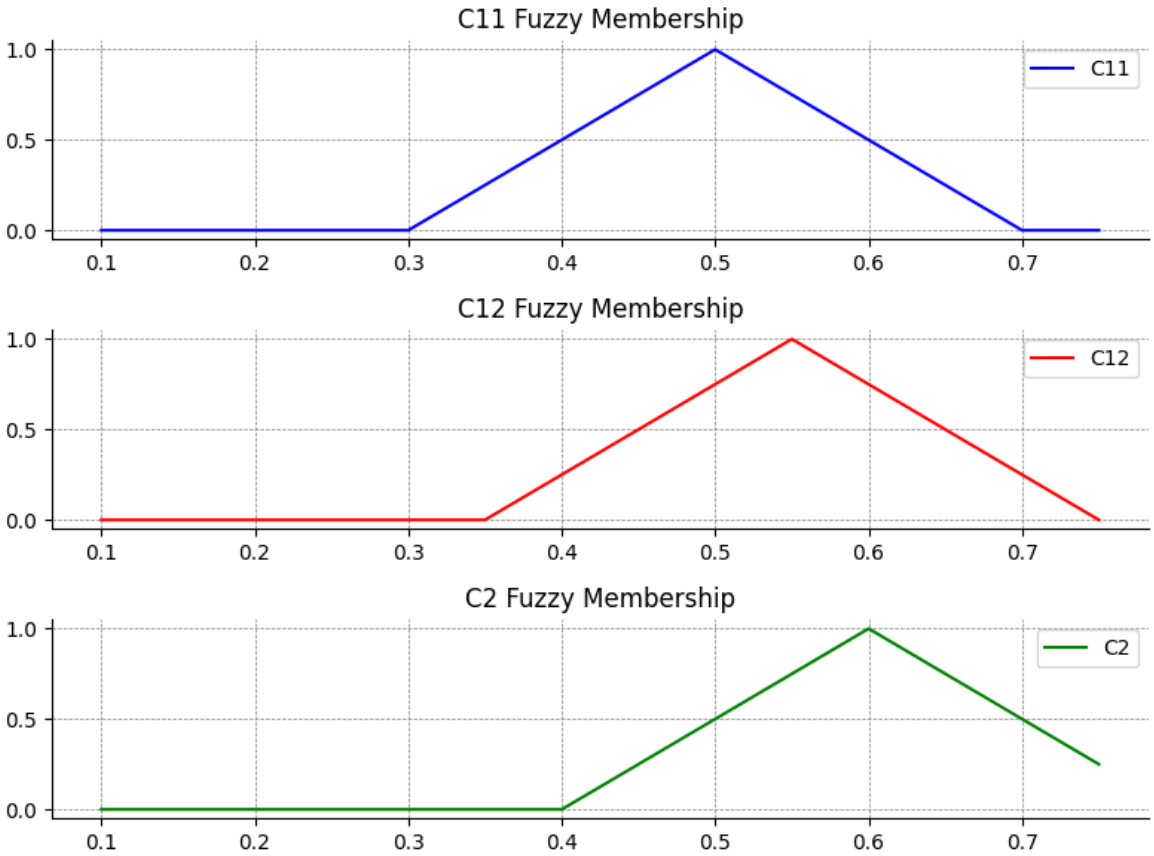

Рисунок 3.3. Трикутні функції належності коефіцієнтів цільової функції

Як і було зазначено, в порівнянні з трапецієподібною нечіткістю, трикутна є більш випадковою, саме тому є тільки одна з 2669 комбінацій з оцінкою ризику (ступенем належності), рівною 1.

Таблиця 3.2

	C11	C12	C2	Optimal Solut	Chance	x_values
1	0.5	0.55	0.6	11.84	1.00	[109.48905109 113.13868613 69.34306569]
2	0.55	0.55	0.55	11.84	0.75	[109.48905109 113.13868613 69.34306569]
3	0.45	0.6	0.6	12.92	0.75	[109.48905109 113.13868613 69.34306569]
4	0.45	0.6	0.55	12.92	0.75	[109.48905109 113.13868613 69.34306569]
5	0.55	0.55	0.6	11.84	0.75	[109.48905109 113.13868613 69.34306569]
6	0.55	0.55	0.65	11.84	0.75	[109.48905109 113.13868613 69.34306569]
7	0.5	0.5	0.55	10.77	0.75	[109.48905109 113.13868613 69.34306569]
8	0.55	0.6	0.65	12.92	0.75	[109.48905109 113.13868613 69.34306569]
9	0.55	0.6	0.6	12.92	0.75	[109.48905109 113.13868613 69.34306569]
10	0.5	0.5	0.65	10.77	0.75	[109.48905109 113.13868613 69.34306569]
11	0.45	0.55	0.65	11.84	0.75	[109.48905109 113.13868613 69.34306569]
12	0.45	0.55	0.6	11.84	0.75	[109.48905109 113.13868613 69.34306569]
13	0.45	0.55	0.55	11.84	0.75	[109.48905109 113.13868613 69.34306569]
14	0.45	0.5	0.55	10.77	0.75	[109.48905109 113.13868613 69.34306569]
15	0.45	0.5	0.6	10.77	0.75	[109.48905109 113.13868613 69.34306569]
16	0.45	0.5	0.65	10.77	0.75	[109.48905109 113.13868613 69.34306569]
17	0.5	0.6	0.65	12.92	0.75	[109.48905109 113.13868613 69.34306569]
18	0.5	0.6	0.6	12.92	0.75	[109.48905109 113.13868613 69.34306569]
19	0.5	0.6	0.55	12.92	0.75	[109.48905109 113.13868613 69.34306569]
20	0.55	0.6	0.55	12.92	0.75	[109.48905109 113.13868613 69.34306569]
21	0.5	0.55	0.65	11.84	0.75	[109.48905109 113.13868613 69.34306569]
22	0.45	0.6	0.65	12.92	0.75	[109.48905109 113.13868613 69.34306569]
23	0.5	0.5	0.6	10.77	0.75	[109.48905109 113.13868613 69.34306569]
24	0.5	0.55	0.55	11.84	0.75	[109.48905109 113.13868613 69.34306569]
25	0.55	0.5	0.55	10.77	0.75	[109.48905109 113.13868613 69.34306569]
26	0.55	0.5	0.65	10.77	0.75	[109.48905109 113.13868613 69.34306569]
27	0.55	0.5	0.6	10.77	0.75	[109.48905109 113.13868613 69.34306569]
28	0.6	0.45	0.55	9.69	0.50	[109.48905109 113.13868613 69.34306569]
29	0.4	0.6	0.5	15.00	0.50	[104.65116279 174.41860465 0.]
30	0.4	0.6	0.55	12.92	0.50	[109.48905109 113.13868613 69.34306569]

3.4 Нечіткі параметри цільової функції з дзвоноподібними функціями належності

Дзвоноподібні функції приналежності, такі як гауссова (1.14) і сигмоїдальна (1.15), забезпечують додаткову гнучкість при моделюванні нечіткості в задачах оптимізації. Ці функції можуть бути особливо корисними під час роботи з комлексними або високорозмірними даними, де нечіткість і варіативність можуть бути неоднорідними та змінюватися в широкому діапазоні.

Гауссова функція належності може бути корисною при моделюванні ситуацій, коли центральні значення є найбільш імовірними, а значення, що відхиляються від середнього, менш імовірні. Функція Гаусса має гладкий, симетричний профіль, який ідеально підходить для представлення таких ситуацій.

Сигмоїдальна функція належності може бути корисною при моделюванні ситуацій, коли потрібно розрізняти дві категорії або стани. Це може бути особливо корисним у вирішенні проблем бінарної оптимізації.

На відміну від трапецієподібних і трикутних функцій, які часто використовуються для представлення невеликих і відносно однорідних неоднозначностей, гауссова та сигмоїдальна функції забезпечують більшу гнучкість і точність у зображенні більш складних або розподілених нечітких сценаріїв.

У першу чергу впровадимо гауссову функцію в параметри цільової функції задачі (2.3) c_{11}, c_{12} і c_2. З цією метою будемо використовувати функцію *fuzz.gaussmf()* з бібліотеки *scikit-fuzzy*. Тут необхідно задати два значення, які будуть відповідати за центр гауссової кривої та за її ширину. Для прикладу, оберемо такі значення: для c_{11}: 0.4 та 0.6; для c_{12}: 0.5 та 0.7; і, нарешті, для c_2: 0.3 та 0.8. Значення трьох параметрів варіюють від 0.1 до 0.8 з кроком 0.05. В коді це записано так:

```
C11_domain = np.arange(0.1, 0.8, 0.05)
C12_domain = np.arange(0.1, 0.8, 0.05)
C2_domain = np.arange(0.1, 0.8, 0.05)

C11_values = fuzz.gaussmf(C11_domain, 0.4, 0.6)
C12_values = fuzz.gaussmf(C12_domain, 0.5, 0.7)
C2_values = fuzz.gaussmf(C2_domain, 0.3, 0.8)
```

На рис. 3.4 продемонстровано, як невизначеність впливає на відповідні параметри.

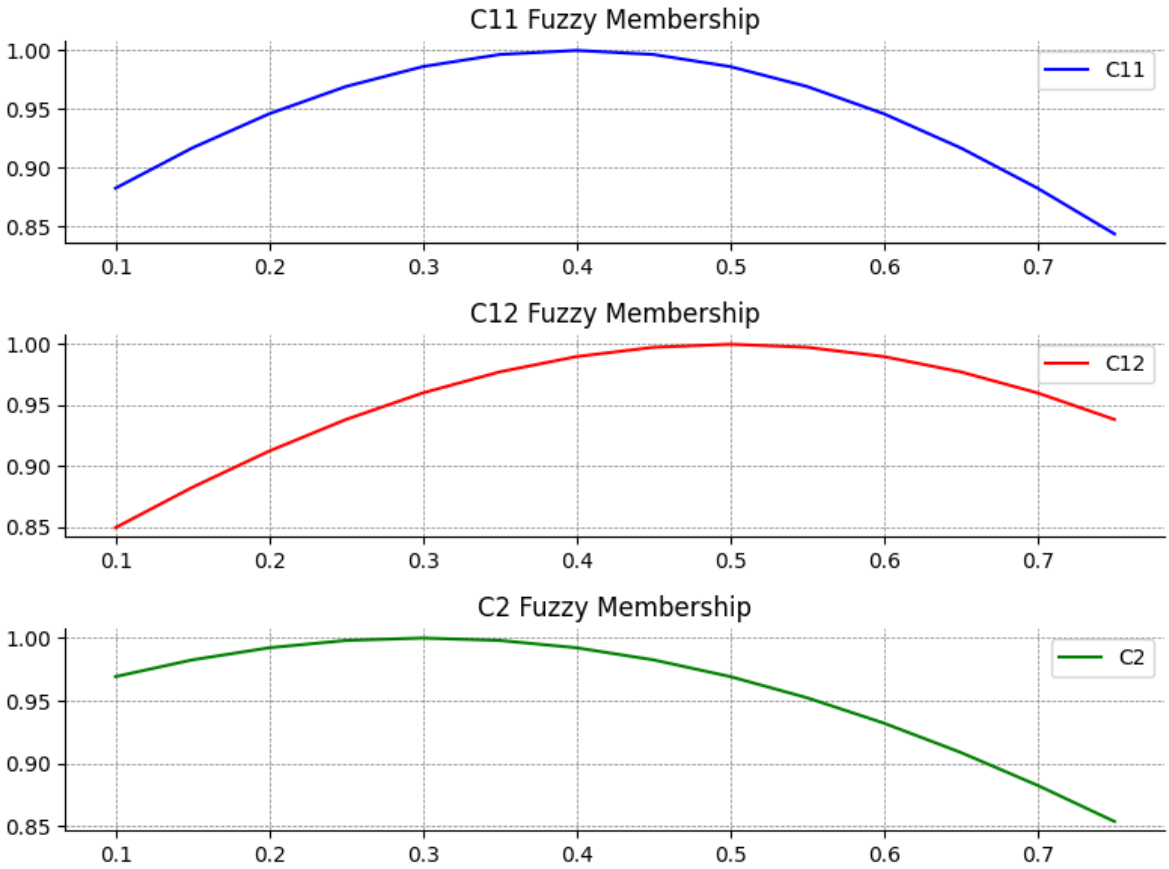

Рисунок 3.4. Гауссові функції належності коефіцієнтів цільової функції задачі (2.3)

Після ітерації всіх можливих комбінацій параметрів c_{11}, c_{12} та c_2 і вирішення оптимізаційної задачі, код генерує відповідну таблицю (таблиця 3.3).

Проаналізувавши таблицю 3.3, ми бачимо, що оцінка ризику є «плавною». Оцінка ризику для 2669 комбінацій варіюється від 0.84 до 1 та не має кроку, більшого за 0.01.

Таблиця 3.3

	C11	C12	C2	Optimal Solut	Chance	x_values
1	0.35	0.55	0.25	27.56	1.00	[104.65116279 174.41860465 0.]
2	0.4	0.5	0.25	23.55	1.00	[104.65116279 174.41860465 0.]
3	0.45	0.5	0.3	20.23	1.00	[104.65116279 174.41860465 0.]
4	0.4	0.55	0.35	20.93	1.00	[104.65116279 174.41860465 0.]
5	0.4	0.5	0.3	20.23	1.00	[104.65116279 174.41860465 0.]
6	0.45	0.45	0.3	16.22	1.00	[104.65116279 174.41860465 0.]
7	0.35	0.5	0.35	16.92	1.00	[104.65116279 174.41860465 0.]
8	0.35	0.55	0.3	24.24	1.00	[104.65116279 174.41860465 0.]
9	0.4	0.45	0.3	16.22	1.00	[104.65116279 174.41860465 0.]
10	0.45	0.5	0.25	23.55	1.00	[104.65116279 174.41860465 0.]
11	0.4	0.45	0.35	12.91	1.00	[104.65116279 174.41860465 0.]
12	0.45	0.45	0.25	19.53	1.00	[104.65116279 174.41860465 0.]
13	0.45	0.5	0.35	16.92	1.00	[104.65116279 174.41860465 0.]
14	0.35	0.45	0.35	12.91	1.00	[104.65116279 174.41860465 0.]
15	0.45	0.45	0.35	12.91	1.00	[104.65116279 174.41860465 0.]
16	0.4	0.55	0.25	27.56	1.00	[104.65116279 174.41860465 0.]
17	0.35	0.45	0.3	16.22	1.00	[104.65116279 174.41860465 0.]
18	0.35	0.55	0.35	20.93	1.00	[104.65116279 174.41860465 0.]
19	0.4	0.55	0.3	24.24	1.00	[104.65116279 174.41860465 0.]
20	0.35	0.5	0.25	23.55	1.00	[104.65116279 174.41860465 0.]
21	0.35	0.45	0.25	19.53	1.00	[104.65116279 174.41860465 0.]
22	0.4	0.5	0.35	16.92	1.00	[104.65116279 174.41860465 0.]
23	0.4	0.45	0.25	19.53	1.00	[104.65116279 174.41860465 0.]
24	0.35	0.5	0.3	20.23	1.00	[104.65116279 174.41860465 0.]
25	0.45	0.55	0.35	20.93	1.00	[104.65116279 174.41860465 0.]
26	0.45	0.55	0.3	24.24	1.00	[104.65116279 174.41860465 0.]
27	0.45	0.55	0.25	27.56	1.00	[104.65116279 174.41860465 0.]
28	0.3	0.6	0.3	28.26	0.99	[104.65116279 174.41860465 0.]
29	0.35	0.55	0.4	17.62	0.99	[104.65116279 174.41860465 0.]
30	0.35	0.6	0.25	31.57	0.99	[104.65116279 174.41860465 0.]

Наступним кроком впровадимо сигмоїдну функцію належності в параметри c_{11}, c_{12} і c_2. Для цього ми будемо використовувати функцію *fuzz.sigmf()* з бібліотеки *scikit-fuzzy*. Функції *fuzz.sigmf()* потрібно надати значення яке визначає, наскільки швидко функція переходить від мінімуму до максимуму та значення, яке визначає «центр» кривої, тобто значення параметрів c_{11}, c_{12} та c_2, при яких відповідна функція досягає свого середнього значення. Визначимо ці значення так: c_{11}: 30 та 0.3; c_{12}: 30 та 0.45 і c_2: 30 та 0.5. Значення трьох параметрів варіюють від 0.1 до 0.8 з кроком 0.05. В коді це реалізовано так:

```
C11_domain = np.arange(0.1, 0.8, 0.05)
C12_domain = np.arange(0.1, 0.8, 0.05)
C2_domain = np.arange(0.1, 0.8, 0.05)

C11_values = fuzz.sigmf(C11_domain, 0.3, 30)
C12_values = fuzz.sigmf(C12_domain, 0.45, 30)
C2_values = fuzz.sigmf(C2_domain, 0.5, 30)
```

На рис. 3.5 продемонстровано, як нечіткість впливає на параметри коефіцієнтів цільової функції.

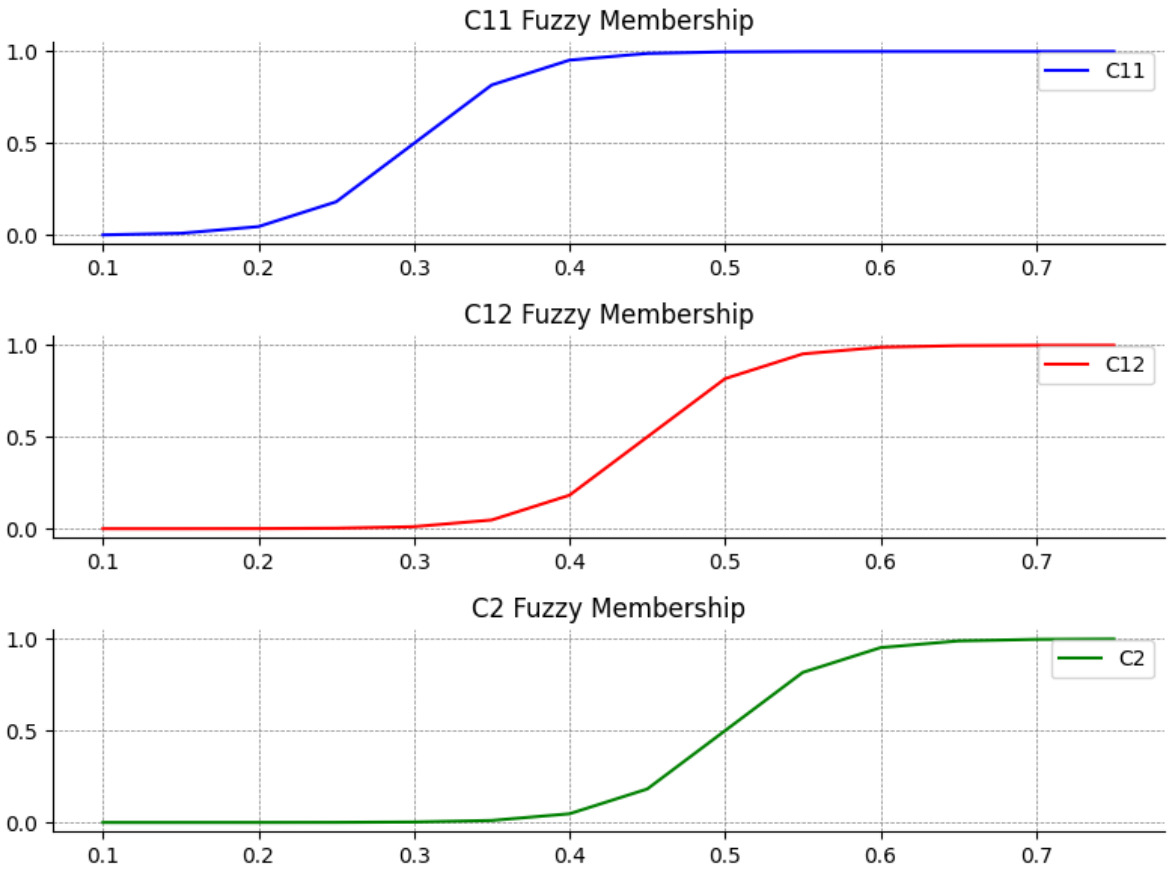

Рисунок 3.5. Сигмоїдні функції належності коефіцієнтів c_{11}, c_{12} та c_2

Після ітерації всіх можливих комбінацій параметрів c_{11}, c_{12} та c_2 і розв'язку оптимізаційної задачі (2.3), код генерує інформацію у відповідну таблицю (табл. 3.4).

Проаналізувавши таблицю 3.4, можна бачити, що значення оцінки ризику «плавно» в усіх 2669 комбінаціях переходить від 0 до 1, що дає нам можливість отримати більш точну оцінку ризику.

Таблиця 3.4

	C11	C12	C2	Optimal Solut	Chance	x_values
1	0.75	0.75	0.75	16.15	1.00	[109.48905109 113.13868613 69.34306569]
2	0.75	0.65	0.75	14.00	1.00	[109.48905109 113.13868613 69.34306569]
3	0.55	0.65	0.75	14.00	1.00	[109.48905109 113.13868613 69.34306569]
4	0.6	0.75	0.75	16.15	1.00	[109.48905109 113.13868613 69.34306569]
5	0.65	0.7	0.7	15.07	1.00	[109.48905109 113.13868613 69.34306569]
6	0.65	0.7	0.75	15.07	1.00	[109.48905109 113.13868613 69.34306569]
7	0.5	0.75	0.75	16.15	1.00	[109.48905109 113.13868613 69.34306569]
8	0.7	0.65	0.7	14.00	1.00	[109.48905109 113.13868613 69.34306569]
9	0.7	0.65	0.75	14.00	1.00	[109.48905109 113.13868613 69.34306569]
10	0.55	0.7	0.7	15.07	1.00	[109.48905109 113.13868613 69.34306569]
11	0.55	0.7	0.75	15.07	1.00	[109.48905109 113.13868613 69.34306569]
12	0.7	0.7	0.7	15.07	1.00	[109.48905109 113.13868613 69.34306569]
13	0.7	0.7	0.75	15.07	1.00	[109.48905109 113.13868613 69.34306569]
14	0.7	0.75	0.7	16.15	1.00	[109.48905109 113.13868613 69.34306569]
15	0.7	0.75	0.75	16.15	1.00	[109.48905109 113.13868613 69.34306569]
16	0.6	0.7	0.75	15.07	1.00	[109.48905109 113.13868613 69.34306569]
17	0.6	0.7	0.7	15.07	1.00	[109.48905109 113.13868613 69.34306569]
18	0.6	0.65	0.7	14.00	1.00	[109.48905109 113.13868613 69.34306569]
19	0.6	0.65	0.75	14.00	1.00	[109.48905109 113.13868613 69.34306569]
20	0.5	0.65	0.7	14.00	1.00	[109.48905109 113.13868613 69.34306569]
21	0.5	0.65	0.75	14.00	1.00	[109.48905109 113.13868613 69.34306569]
22	0.55	0.75	0.7	16.15	1.00	[109.48905109 113.13868613 69.34306569]
23	0.55	0.75	0.75	16.15	1.00	[109.48905109 113.13868613 69.34306569]
24	0.65	0.75	0.7	16.15	1.00	[109.48905109 113.13868613 69.34306569]
25	0.5	0.7	0.7	15.07	1.00	[109.48905109 113.13868613 69.34306569]
26	0.5	0.7	0.75	15.07	1.00	[109.48905109 113.13868613 69.34306569]
27	0.65	0.75	0.75	16.15	1.00	[109.48905109 113.13868613 69.34306569]
28	0.75	0.65	0.7	14.00	1.00	[109.48905109 113.13868613 69.34306569]
29	0.5	0.75	0.7	16.15	1.00	[109.48905109 113.13868613 69.34306569]
30	0.75	0.7	0.75	15.07	1.00	[109.48905109 113.13868613 69.34306569]

ВИСНОВКИ

У ході виконання дипломної роботи було виконано наступні завдання:

- Проведено теоретичне дослідження еколого-економічної моделі Леонтьєва-Форда. Ця модель дозволяє моделювати взаємозв’язки між різними секторами економіки, при цьому враховуючи екологічні обмеження, що робить її надзвичайно цінною в контексті проблем сучасного світу. Було розглянуто оптимізаційну постановку у вигляді задачі лінійного програмування, де модель Леонтьєва-Форда є основною частиною обмежень.
- Було інтерпретовано двовимірну оптимізаційну модель Леонтьєва-Форда в коді мовою *Python*. Цей код вираховує оптимальні значення цільової функції та невідомих змінних, дозволяючи зрозуміти, як взаємодіють різні фактори в рамках моделі.
- В роботі було опрацьовано тривимірний варіант моделі. Цей процес був записаний та продемонстрований. Для цього варіанту також було написано код.
- Для коефіцієнтів цільової функції оптимізаційної задачі було розглянуто варіанти трапецієподібної, трикутної, гауссової та сигмоїдної нечіткості, що поскладнило задачу, але також дало більше гнучкості щодо можливих варіантів оптимального вибору. Був написаний код, який оптимізує модель для кожної допустимої комбінації нечітких параметрів.

Представлення результатів моделювання у вигляді таблиці надає можливість особі, що приймає рішення (ОПР) здійснювати оптимальний вибір у ситуації з нечіткими параметрами цільової функції, спираючись на відповідний масив інформації. Зокрема, обирати компроміс між кращим значенням критерію та меншим значенням міри належності.

Запропонована в роботі методика та результати дослідження підтверджують важливість розробки і використання математичних моделей, зокрема моделі Леонтьєва-Форда, в еколого-економічній проблематиці, для кращого розуміння та прийняття відповідних управлінських рішень щодо оптимального використання світових ресурсів. Також варто зауважити, що використані інструменти дозволяють працювати з реальними даними, які часто містять нечіткість та невизначеність Можливо, результати, які отримано в роботі, будуть корисними для майбутніх досліджень і практичних застосувань.

СПИСОК ДЖЕРЕЛ ПОСИЛАННЯ

1. Ляшенко І.М. Моделювання економічних, екологічних і соціальних процесів: навч. посіб. / І.М. Ляшенко, М.В. Коробова, І.А. Горіцина. – К.: Видавничо-поліграфічний центр «Київський університет», 2010. – 320 с.
2. Ляшенко І.М. Економіко-математичні методи та моделі сталого розвитку – Київ: Вища школа, 1999. – 236 с.
3. Klir G.J., Yuan B. Fuzzy Sets and Fuzzy Logic: Theory and Applications – New York: Prentice Hall, 1995. – 592 p.
4. McKinney W. Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython – Sebastopol: O'Reilly Media, 2018. – 544 p.
5. Leontief W. Input-Output Economics – New York: Oxford University Press, 1986. – 436 p.

ДОДАТОК А

```
from scipy.optimize import linprog
import matplotlib.pyplot as plt
import numpy as np

A11 = 0.3
A12 = 0.4
A21 = 0.5
A22 = 0.6
C1 = 0.9
C2 = 0.8
B1 = 4
B2 = 4
b = 80
E1=E2=1

c = [-C1*(E1-A11)+C2*A21, C2*(E2-A22)-C1*A12]

A_ub = [[B1, B2],
        [-E1+A11, -A12],
        [-A21,-E2+A22]]
b_ub = [b,0,0]

x0_bounds = (0, None)
x1_bounds = (0, None)

res = linprog(c, A_ub=A_ub, b_ub=b_ub, bounds=[x0_bounds, x1_bounds],
method='highs')

print('Отпимальне значення:', -res.fun, '\nX1, X2:', res.x)
```

```
x1_values = np.linspace(0, 100, 400)
x2_values = np.linspace(0, 100, 400)

constraint1 = (1 - A11) * x1_values / A12
constraint2 = A21 * x1_values / (1 - A22)
constraint3 = (b - B1 * x1_values) / B2

plt.figure(figsize=(10,10))

plt.plot(x1_values, constraint1, label='(E1 - A11) * x1 - A12 * x2 >= 0')
plt.fill_between(x1_values, 0, constraint1, where=(x2_values<=constraint1),
alpha=0.1, color='red')

plt.plot(x1_values, constraint2, label='A21 * x1 - (E2 - A22) * x2 >= 0')
plt.fill_between(x1_values, 0, constraint2, where=(x2_values<=constraint2),
alpha=0.1, color='blue')

plt.plot(x1_values, constraint3, label='B1 * x1 + B2 * x2 <= b')
plt.fill_between(x1_values, constraint3, 0, alpha=0.1, color='green')

plt.plot(res.x[0], res.x[1], 'ro')

plt.xlim(0, 50)
plt.ylim(0, 50)
plt.xlabel('x1')
plt.ylabel('x2')
plt.legend()
plt.grid(True)
plt.show()
```

ДОДАТОК Б

```
import numpy as np
from scipy.optimize import linprog
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

A11 = np.array([[0.5, 0.3], [0.4, 0.3]])
A12 = np.array([[0.3], [0.2]])
A21 = np.array([[0.3, 0.2]])
A22 = np.array([[0.2]])

b1 = np.array([50])
b2 = np.array([150])

B11 = np.array([0.6])
B12 = np.array([0.5])
B2 = np.array([0.4])
E1 = E2 = 1

C11=0.5
C12=0.6
C2=0.55

c_coeff = [-(C11 * (1 - A11[0][0]) - C12 * A11[1][0] - C2*A21[0][0]),
          -(-C11*A11[0][1]+C12*(1-A11[1][1])-C2*A21[0][1]),
          -(-C11*A12[0]-C12*A12[1]-C2*(A22-1))]

A_ub = [[B11[0], B12[0], B2[0]],
       [-1 + A11[0][0], A11[0][1],  A12[0]],
       [A11[1][0], -1 + A11[1][1], A12[1]],
```

```
        [-A21[0][0], -A21[0][1], -A22+1]]
b_vector = [b2[0], 0, 0, 0]
x_bounds = [(0, None), (0, None), (0, None)]

res = linprog(c_coeff, A_ub=A_ub, b_ub=b_vector, bounds=x_bounds,
method='highs')
y11=(1-A11[0][0])*res.x[0]-A11[0][1]*res.x[1]-A12[0]*res.x[2]
y12=-A11[1][0]*res.x[0]+(1-A11[1][1])*res.x[1]-A12[1]*res.x[2]
y2=A21[0][0]*res.x[0]+A21[0][1]*res.x[1]+(A22-1)*res.x[2]
b=B11[0]*res.x[0]+B12[0]*res.x[1]+B2[0]*res.x[2]

if res.success and y11>=0 and y12>=0 and y2>=0 and b>=b2[0]:
    print("Optimal Solution=", -res.fun, "x_values=", res.x)

fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')

ax.scatter(res.x[0], res.x[1], res.x[2], c='r', marker='o')
x11_grid, x12_grid = np.meshgrid(np.linspace(0, 300, 300),
                                 np.linspace(0, 300, 300))

x2_grid1 = ((1 - A11[0][0]) * x11_grid - A11[0][1] * x12_grid) / A12[0]
x2_grid2 = (-A11[1][0] * x11_grid + (1 - A11[1][1]) * x12_grid) / A12[1]
x2_grid3 = (A21[0][0] * x11_grid + A21[0][1] * x12_grid + (A22 - 1) * x2_grid2)
x2_grid4 = (b2[0] - B11[0]*x11_grid - B12[0]*x12_grid) / B2[0]

x2_grid1[x2_grid1 > 200] = np.nan
x2_grid2[x2_grid2 > 200] = np.nan
x2_grid3[x2_grid3 > 200] = np.nan
x2_grid4[x2_grid4 > 200] = np.nan
```

```
x2_grid1[0 > x2_grid1] = np.nan
x2_grid2[0 > x2_grid2] = np.nan
x2_grid3[0 > x2_grid3] = np.nan
x2_grid4[0 > x2_grid4] = np.nan

ax.plot_surface(x11_grid, x12_grid, x2_grid1, color='b', alpha=0.5, rstride=100,
cstride=100)
ax.plot_surface(x11_grid, x12_grid, x2_grid2, color='y', alpha=0.5, rstride=100,
cstride=100)
ax.plot_surface(x11_grid, x12_grid, x2_grid3, color='g', alpha=0.5, rstride=100,
cstride=100)
ax.plot_surface(x11_grid, x12_grid, x2_grid4, color='r', alpha=0.5, rstride=100,
cstride=100)

ax.set_xlabel('X11')
ax.set_ylabel('X12')
ax.set_zlabel('X2')
ax.set_zlim([0, 200])
plt.xlim(0, 200)
plt.ylim(0, 200)
plt.show()
```

ДОДАТОК В

```
import numpy as np
from scipy.optimize import linprog
import skfuzzy as fuzz
import pandas as pd
import tkinter as tk
from pandastable import Table
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

A11 = np.array([[0.5, 0.3], [0.4, 0.3]])
A12 = np.array([[0.3], [0.2]])
A21 = np.array([[0.3, 0.2]])
A22 = np.array([[0.2]])

b1 = np.array([50])
b2 = np.array([150])

B11 = np.array([0.6])
B12 = np.array([0.5])
B2 = np.array([0.4])
E1 = E2 = 1

C11_domain = np.arange(0.1, 0.8, 0.05)
C12_domain = np.arange(0.1, 0.8, 0.05)
C2_domain = np.arange(0.1, 0.8, 0.05)

C11_values = fuzz.trapmf(C11_domain, [0.3, 0.4, 0.6, 0.85])
C12_values = fuzz.trapmf(C12_domain, [0.35, 0.45, 0.7, 0.95])
C2_values = fuzz.trapmf(C2_domain, [0.4, 0.5, 0.6, 0.8])
```

```
data_list = []
x11_values = []
x12_values = []
x2_values = []

for i in range(len(C11_domain)):
    for j in range(len(C12_domain)):
        for k in range(len(C2_domain)):

            C11 = C11_domain[i]
            C12 = C12_domain[j]
            C2 = C2_domain[k]

            c_coeff = [-(C11_domain[i] * (1 - A11[0][0]) - C12_domain[j] * A11[1][0]
- C2_domain[k]*A21[0][0]),
                       -(-C11_domain[i]*A11[0][1]+C12_domain[j]*(1-A11[1][1])-
C2_domain[k]*A21[0][1]),
                       -(-C11_domain[i]*A12[0]-C12_domain[j]*A12[1]-
C2_domain[k]*(A22-1))]

            A_ub = [[B11[0], B12[0], B2[0]],
                    [-1 + A11[0][0], A11[0][1],  A12[0]],
                    [A11[1][0], -1 + A11[1][1], A12[1]],
                    [-A21[0][0], -A21[0][1], -A22+1]]

            b_vector = [b2[0], 0, 0, 0]

            x_bounds = [(0, None), (0, None), (0, None)]
```

```
        res = linprog(c_coeff, A_ub=A_ub, b_ub=b_vector, bounds=x_bounds,
method='highs')

        y11=(1-A11[0][0])*res.x[0]-A11[0][1]*res.x[1]-A12[0]*res.x[2]
        y12=-A11[1][0]*res.x[0]+(1-A11[1][1])*res.x[1]-A12[1]*res.x[2]
        y2=A21[0][0]*res.x[0]+A21[0][1]*res.x[1]+(A22-1)*res.x[2]
        b=B11[0]*res.x[0]+B12[0]*res.x[1]+B2[0]*res.x[2]

        if res.success and y11>=0 and y12>=0 and y2>=0 and b>=b2[0]:
            y22=A21[0][0]*res.x[0]+A21[0][1]*res.x[1]+(A22-1)*res.x[2]
            data_list.append({"C11": C11,
                              "C12": C12,
                              "C2": C2,
                              "Optimal Solution": -res.fun,
                              "Chance": "{:.2f}".format(min(C11_values[i],
C12_values[j], C2_values[k])),
                              "x_values": res.x})

            x11_values.append(res.x[0])
            x12_values.append(res.x[1])
            x2_values.append(res.x[2])

df = pd.DataFrame(data_list)

root = tk.Tk()
frame = tk.Frame(root)
frame.pack(fill='both', expand=True)
pt = Table(frame, dataframe=df, showtoolbar=True, showstatusbar=True)
pt.show()
```

```
fig, ax = plt.subplots(nrows=3, figsize=(8, 6))

ax[0].plot(np.arange(0.1, 0.8, 0.05), C11_values, 'b', linewidth=1.5, label='C11')
ax[0].set_title('C11 Fuzzy Membership')
ax[0].legend()

ax[1].plot(np.arange(0.1, 0.8, 0.05), C12_values, 'r', linewidth=1.5, label='C12')
ax[1].set_title('C12 Fuzzy Membership')
ax[1].legend()

ax[2].plot(np.arange(0.1, 0.8, 0.05), C2_values, 'g', linewidth=1.5, label='C2')
ax[2].set_title('C2 Fuzzy Membership')
ax[2].legend()

for ax in ax.flat:
    ax.spines['top'].set_visible(False)
    ax.spines['right'].set_visible(False)
    ax.grid(color='gray', linestyle='--', linewidth=0.5)

plt.tight_layout()
plt.show()

fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')

ax.scatter(x11_values, x12_values, x2_values, c='r', marker='o')

x11_grid, x12_grid = np.meshgrid(np.linspace(min(x11_values),
max(x11_values), len(x11_values)),
```

```
                    np.linspace(min(x12_values), max(x12_values),
len(x12_values)))

x2_grid1 = ((1 - A11[0][0]) * x11_grid - A11[0][1] * x12_grid) / A12[0]
x2_grid2 = (-A11[1][0] * x11_grid + (1 - A11[1][1]) * x12_grid) / A12[1]
x2_grid3 = (A21[0][0] * x11_grid + A21[0][1] * x12_grid + (A22 - 1) * x2_grid2)
x2_grid4 = (b2[0] - B11[0]*x11_grid - B12[0]*x12_grid) / B2[0]

x2_grid1[x2_grid1 > 300] = np.nan
x2_grid2[x2_grid2 > 300] = np.nan
x2_grid3[x2_grid3 > 300] = np.nan
x2_grid4[x2_grid4 > 300] = np.nan

x2_grid1[0 > x2_grid1] = np.nan
x2_grid2[0 > x2_grid2] = np.nan
x2_grid3[0 > x2_grid3] = np.nan
x2_grid4[0 > x2_grid4] = np.nan

ax.plot_surface(x11_grid, x12_grid, x2_grid1, color='b', alpha=0.5, rstride=100,
cstride=100)
ax.plot_surface(x11_grid, x12_grid, x2_grid2, color='y', alpha=0.5, rstride=100,
cstride=100)
ax.plot_surface(x11_grid, x12_grid, x2_grid3, color='g', alpha=0.5, rstride=100,
cstride=100)
ax.plot_surface(x11_grid, x12_grid, x2_grid4, color='r', alpha=0.5, rstride=100,
cstride=100)

ax.set_xlabel('X11')
ax.set_ylabel('X12')
ax.set_zlabel('X2')
```

```
ax.set_zlim([0, 300])
plt.xlim(0, 300)
plt.ylim(0, 200)
plt.show()

root.mainloop()
```