

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ДНІПРОВСЬКА ПОЛІТЕХНІКА»



ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
Кафедра інформаційних технологій та комп'ютерної інженерії

ТЕОРІЯ ТА МЕТОДИ ПРИЙНЯТТЯ РІШЕНЬ

Методичні рекомендації до виконання лабораторних робіт
для здобувачів ступеня магістра
спеціальностей галузі знань F Інформаційні технології

Дніпро
НТУ «ДП»
2025

Заболотний К.С.

Теорія та методи прийняття рішень [Електронний ресурс] : методичні рекомендації до виконання лабораторних робіт для здобувачів ступеня магістра спеціальностей галузі знань F Інформаційні технології / К.С. Заболотний, А. В. Кожевников, Л. І. Мещеряков, Н.О. Соколова ; М-во освіти і науки України, Нац. техн. ун-т «Дніпровська політехніка». – Дніпро : НТУ «ДП», 2025. – 107 с.

Автори:

К. С. Заболотний, д-р техн. наук, проф.,
А.В. Кожевников, канд. техн. наук, доц.,
Л. І. Мещеряков, д-р техн. наук, проф.,
Н.О. Соколова, канд. техн. наук, доц.

Затверджено кафедрою інформаційних технологій та комп'ютерної інженерії (протокол № 6 від 09.10.2025 р.).

Уміщено теоретичні відомості за темами лекційного курсу, варіанти лабораторних завдань з рекомендаціями до їх виконання, контрольні питання, список використаної та рекомендованої літератури.

Зміст методичних рекомендацій відповідає програмі вибіркової навчальної дисципліни «Теорія та методи прийняття рішень» і адресовано магістрам, які проходять підготовку за спеціальностями галузі знань F Інформаційні технології, а також здобувачам усіх інших спеціальностей, які можуть обрати зазначену дисципліну для вивчення в межах формування індивідуальної траєкторії навчання.

Відповідальний за випуск завідувач кафедри інформаційних технологій та комп'ютерної інженерії В. В. Гнатушенко, д-р техн. наук, проф.

1. ЗАГАЛЬНІ ПОЛОЖЕННЯ

Дисципліна “Теорія та методи прийняття рішень” – вибіркова складова фахової підготовки магістрів усіх освітньо-професійних програм спеціальностей факультету інформаційних технологій. Мета дисципліни – формування у здобувачів вищої освіти умінь та компетентностей щодо методів прийняття рішень та програмних засобів їх підтримки.

Методичні рекомендації призначені для закріплення теоретичних знань, набутих здобувачами вищої освіти в лекційному курсі, а також формування навичок виконання лабораторних робіт щодо методів прийняття рішень та програмних засобів їх підтримки.

У підсумку виконання лабораторних робіт здобувачі вищої освіти – майбутні фахівці повинні отримати результати навчання у відповідності з вимогами робочої програми дисципліни, які представлені в табл. 1.1.

Таблиця 1.1. Дисциплінарні результати навчання, що отримуються у підсумку вивчення дисципліни “Теорія та методи прийняття рішень”

Дисциплінарні результати навчання (ДРН)	
шифр ДРН	зміст
ДРН-1	Класифікувати задачі прийняття рішень та оптимізації і обирати відповідні методи щодо їх розв'язання
ДРН-2	Розв'язувати задачі лінійного, нелінійного програмування та транспортні задачі
ДРН-3	Знати основні підходи статистичної теорії прийняття рішень та теорії парних ігор. Вміти обирати критерії прийняття рішень.
ДРН-4	Застосовувати методи штучного інтелекту для рішення задач класифікації даних
ДРН-5	Розробляти та використовувати програмне забезпечення для обробки і візуалізації багатовимірних даних

Мета лабораторних робіт – поглибити і систематизувати набуті здобувачами вищої освіти на лекціях теоретичні знання з дисципліни “Теорія та методи прийняття рішень” та сформувати у майбутніх фахівців з інформаційних систем та технологій професійні компетентності (знання, вміння і навички) використання методів прийняття рішень та програмних засобів їх підтримки.

Лабораторні роботи передбачають застосування авторського та вільно розповсюдженого програмного забезпечення. Кожна лабораторна робота має назву, ціль, постановку задачі, контрольні питання і завдання, рекомендації щодо виконання роботи та вимоги до оформлення звіту, а також варіанти індивідуальних завдань.

2. ПОСТАНОВКА ЗАДАЧ ЛАБОРАТОРНИХ РОБІТ, МЕТОДИЧНІ РЕКОМЕНДАЦІЇ ДЛЯ ЇХ ВИКОНАННЯ ТА ІНДИВІДУАЛЬНІ ЗАВДАННЯ

2.1. Лабораторна робота № 1

Прийняття рішень за допомогою методів лінійного програмування

Об'єкт – завдання лінійного програмування. Предмет – завдання лінійного програмування з двома вільними змінними. Мета – ознайомлення з методами прийняття рішень за допомогою апарату лінійного програмування.

Стислі теоретичні відомості

Основне завдання лінійного програмування формулюється таким чином: знайти невід'ємні значення змінних x_j ($j=1..n$) – елементів рішення, які задовольняють обмеженням у вигляді системи рівнянь

$$\begin{cases} a_{11}x_1 + \dots + a_{1j}x_j + \dots + a_{1n}x_n = b_1 \\ : \\ a_{m1}x_1 + \dots + a_{mj}x_j + \dots + a_{mn}x_n = b_m \end{cases}, \quad (2.1.1)$$

або в векторно-матричному вигляді

$$A \cdot \vec{x} = \vec{b}, \quad (2.1.2)$$

де $A = \{a_{ij}\}_{i=1..m; j=1..n}$ – матриця системи лінійних обмежень у вигляді рівнянь, $\vec{b} = \{b_i\}_{i=1..m}$ – вектор вільних членів, $\{x_j\}_{j=1..n}$ – вектор невідомих. При цьому повинна мінімізуватися цільова функція

$$L(\vec{x}) = c_1x_1 + \dots + c_nx_n, \quad (2.1.3)$$

або

$$L(\vec{x}) = c^T \cdot \vec{x}. \quad (2.1.4)$$

Будь-яке завдання лінійного програмування можна звести до канонічного вигляду. Випадки, коли потрібно максимізувати цільову функцію, зводяться до канонічного вигляду шляхом заміни: $L' = -L$. Випадки, коли одне або кілька накладених обмежень мають вигляд нерівностей $a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n \leq b_i$ або $a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n \geq b_i$, зводяться до канонічного вигляду шляхом введення

додаткових змінних $y \geq 0$ відповідно як $a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n + y = b_i$ або $a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n - y = b_i$.

Нехай дана прямокутна матриця розміром $m \times n$. Виділимо в ній k довільних рядків і k довільних стовпців. Визначник k -го порядку, складений з елементів матриці, розташованих на перетині виділених стовпців і рядків, називається мінором k -го порядку матриці. Найбільший порядок мінору матриці, відмінного від 0, називається рангом. Відмінний від 0 мінор, порядок якого дорівнює рангу матриці, називається базисним.

Розглянемо систему з m лінійних рівнянь з n невідомими (2.1.1) або (2.1.2). Рішенням цієї системи називається вектор $\vec{x}^*$ з n чисел, що звертають рівняння до тотожностей. Система сумісна, якщо вона має хоча б одне рішення, в іншому випадку вона несумісна. Система визначена, якщо вона сумісна і має єдиний розв'язок, в іншому випадку вона невизначена. Матриця A' , що отримується шляхом додавання стовпця вільних членів до матриці коефіцієнтів, називається розширеною матрицею системи. Для сумісності системи необхідно і достатньо, щоб ранг матриці коефіцієнтів дорівнював рангу розширеної матриці $r(A) = r(A') = r$. У цьому випадку величина r називається рангом системи. Якщо ранг спільної системи дорівнює числу невідомих n , то система визначена. Якщо ранг спільної системи менше n , то система невизначена.

Розглянемо який-небудь базисний мінор матриці A і виділимо в ньому довільний рядок. Елементи цього рядка є коефіцієнтами при r невідомих, ці невідомі називаються базисними, інші вільними. При отриманні рішення невизначеною системи значення m базисних змінних виражається через $n - m$ вільних, які вибираються довільно.

Рішення – це будь-який набір змінних, які відповідають обмеженням (2.1.1).

Припустиме рішення – це рішення з невід'ємними значеннями всіх змінних.

Область припустимих рішень (ОПР) – множина всіх припустимих рішень. У загальному випадку для задач лінійного програмування це множина крапок простору розмірності $n - m$, координати яких є дійсними числами, що обмежена m гіперплощинами. ОПР може бути як нескінченною, так і кінцевою, тобто гіпербагатокутником.

Опорне рішення – припустиме рішення, яке розташовується на межі ОПР.

Оптимальне рішення – опорне рішення, яке мінімізує цільову функцію (2.1.3).

Далі розглядаються методи рішення основного завдання лінійного програмування.

Геометричне рішення найбільш наочно, коли кількість вільних змінних $k = n - m = 2$.

Пронумеруємо змінні таким чином, щоб вільні змінні були x_1, x_2 і виразимо через них базисні. При цьому система обмежень набуде вигляду

$$\begin{cases} x_3 = \alpha_{31}x_1 + \alpha_{32}x_2 + \beta_3 \\ \vdots \\ x_n = \alpha_{n1}x_1 + \alpha_{n2}x_2 + \beta_n \end{cases} \quad (2.1.5)$$

Зобразимо пару вільних змінних у вигляді точки з координатами (x_1, x_2) і врахуємо, що за умовою основного завдання лінійного програмування значення елементів невід'ємні.

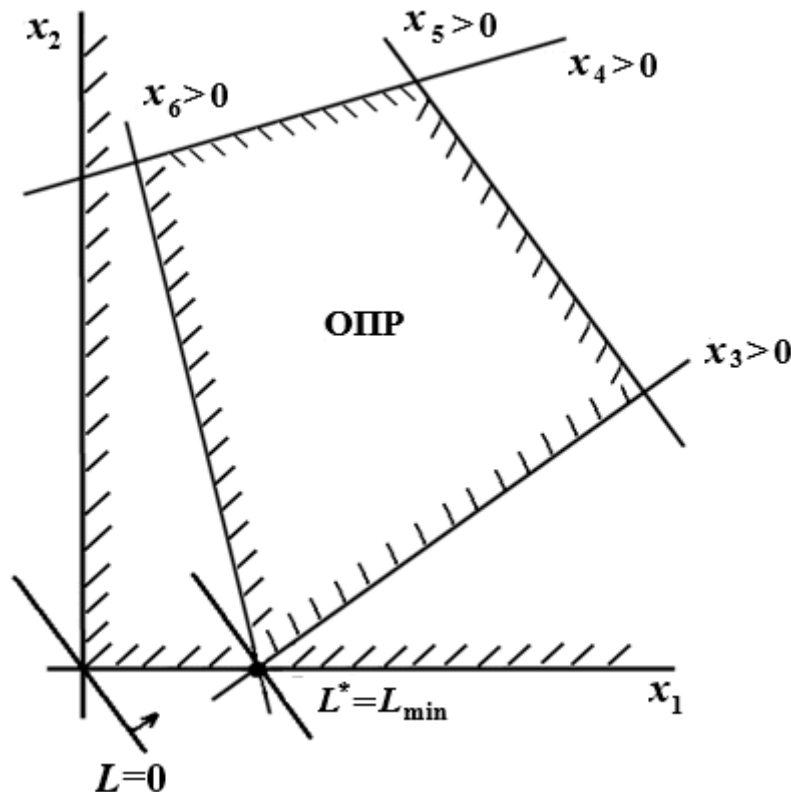


Рис. 2.1.1. Геометричне рішення основного завдання лінійного програмування

Для пошуку оптимального рішення виразимо цільову функцію через вільні змінні

$$L = \gamma_1 x_1 + \gamma_2 x_2 + \gamma_0. \quad (2.1.6)$$

Мінімум функції (2.1.6) збігається з мінімумом функції

$$L' = \gamma_1 x_1 + \gamma_2 x_2, \quad (2.1.7)$$

який досягається в тій же точці. При $L'=0$ рівняння прямої (2.1.7) набуває вигляду $x_2 = -\gamma_1 \cdot x_1 / \gamma_2$. Така пряма зветься опорною і проходить через початок системи координат. При $L' \neq 0$ пряма, що задається рівнянням (2.1.7), буде пересуватися паралельно самій собі. Перший перетин прямої з областю припустимих рішень (ОПР) є оптимальним рішенням. Оптимальне рішення

задачі лінійного програмування досягається на межі ОПР. Можливо, що оптимального рішення не існує; можливо воно єдине; можливо їх декілька, або безліч.

При числі вільних змінних $k > 2$ геометричне рішення втрачає свою наочність. Найбільш поширеним чисельним методом вирішення основного завдання лінійного програмування є симплекс-метод.

Позначимо $\{x_i\}_{i=1,\dots,m}$ – базисні змінні, $\{y_j\}_{j=1,\dots,n-m}$ – вільні. Вирішуючи систему обмежень (2.1.1) щодо базисних змінних, маємо

$$\begin{cases} x_1 = S_{10} - (S_{11}y_1 + \dots + S_{1n-m}y_{n-m}) \\ \vdots \\ x_r = S_{r0} - (S_{r1}y_1 + \dots + S_{rn-m}y_{n-m}) \\ \vdots \\ x_n = S_{n0} - (S_{n1}y_1 + \dots + S_{nn-m}y_{n-m}) \end{cases} \quad (2.1.8)$$

Вважаючи вільні змінні рівними 0, отримуємо базисне рішення $\{x_i\}_{i=1,\dots,m} = \{S_{i0}\}_{i=1,\dots,m}$. Воно є опорним, якщо $\{x_i\}_{i=1,\dots,m} \geq 0$. Якщо ж існує одне або кілька значень r , для яких виконується умова $S_{r0} < 0$, то можливо перерозв'язати систему (2.1.8), вивівши змінну x_r з базису і ввівши замість неї деяку вільну змінну y_k і тим самим домогтися, щоб нове значення вільного члена S'_{r0} було б невід'ємним.

Застосовуючи процедуру обміну вільних і базисних змінних при необхідності багаторазово, в кінці кінців, можна отримати опорне рішення.

Виразимо тепер через поточні вільні змінні цільову функцію:

$$L = S_{m+1\ 0} - (S_{m+1\ 1}y_1 + \dots + S_{m+1\ n-m}y_{n-m}). \quad (2.1.9)$$

Якщо всі коефіцієнти $\{S_{m+1j}\}_{j=1,\dots,n-m} \leq 0$, то рішення коли всі вільні змінні дорівнюють 0, є оптимальним, оскільки, збільшуючи значення вільних змінних, зменшити значення цільової функції неможливо. Якщо ж існує одне або кілька значень k , для яких виконується умова $S_{m+1k} > 0$, то для отримання оптимального рішення знову використовується процедура заміни базисних змінних.

Таблиця вигляду

$$\begin{pmatrix} S_{10} & S_{11} & \dots & S_{1n-m} \\ \vdots & \vdots & & \vdots \\ S_{m0} & S_{m1} & \dots & S_{mn-m} \\ 0 & S_{m+11} & \dots & S_{m+1n-m} \end{pmatrix}. \quad (2.1.10)$$

називається симплекс-таблицею і використовується як зручна форма відображення коефіцієнтів основного завдання лінійного програмування при

заміні базисних змінних. При такій формі запису симплекс-таблиці оптимізується функція $L' = L - S_{m+10}$.

Для отримання відповідних формул перерахунку коефіцієнтів симплекс-таблиці при заміні базисних змінних на вільні виразимо зі співвідношення (2.1.8) вільну змінну через базисну

$$y_k = \frac{1}{S_{rk}} [S_{r0} - (S_{r1}y_1 + \dots + S_{rn-m}y_{n-m}) + x_r] = \frac{1}{S_{rk}} \left[S_{r0} - \sum_{j=1}^{n-m} S_{rj}y_j - x_r \right]. \quad (2.1.11)$$

Розв'язуючи спільно рівняння (2.1.8) і (2.1.11) отримуємо

$$x_i = S_{i0} - \frac{S_{ik} - S_{r0}}{S_{rk}} - \sum_{j=1}^{n-m} \left(S_{ij} - \frac{S_{ik}S_{rj}}{S_{rk}} \right) y_j + \frac{S_{ik}}{S_{rk}} x_r. \quad (2.1.12)$$

Остаточні співвідношення для заміни базисних змінних на вільні мають вигляд:

$$\begin{cases} S'_{ij} = S_{ij} - \frac{S_{ik}S_{rj}}{S_{rk}} & i \neq r, j \neq k \\ S'_{ik} = -\frac{S_{ik}}{S_{rk}} & i \neq r, j = k \text{ (для розрішаючого стовпця)} \\ S'_{rj} = \frac{S_{rj}}{S_{rk}} & i = r, j \neq k \text{ (для розрішаючого рядка)} \\ S'_{rk} = \frac{1}{S_{rk}} & \text{(для розрішаючого елемента)} \end{cases}. \quad (2.1.13)$$

У симплекс-таблиці рядки відповідають базисним змінним, а стовпці – вільним. При заміні вільної і базисної змінної рядок r , що відповідає базисній змінній, що виводиться з базису, носить назву розрішаючого рядка; стовпець k , що відповідає вільній змінній, яка вводиться в базис, носить назву розрішаючого стовпця; а елемент, що лежить на перетині розрішаючого стовпця і розрішаючого рядка, називається розрішаючим елементом.

Алгоритм вирішення основного завдання лінійного програмування симплекс-методом повинен бути доповнений правилом вибору розрішаючого стовпця і розрішаючого рядка.

Правило вибору розрішаючого стовпця і розрішаючого рядка при пошуку опорного рішення має вигляд:

- критерій досягнення опорного рішення $\{S_{i0}\}_{i=1,\dots,m} \geq 0$. Якщо ж є рядок, в якому $S_{r0} < 0$, то член $S_{rk} < 0$ визначає номер розрішаючого стовпця k ;

- у якості розрішаючого рядка обирається рядок r , який має мінімальне додатнє значення відношення $\alpha_r = S_{r0}/S_{rk}$.

Правило вибору розрішаючого стовпця і розрішаючого рядка при пошуку оптимального рішення має вигляд:

- критерій досягнення оптимального рішення $\{S_{m+1j}\}_{j=1,\dots,n-m} \leq 0$ (при пошуку мінімуму) або $\{S_{m+1j}\}_{j=1,\dots,n-m} \geq 0$ (при пошуку максимуму). У якості розрішаючого обирається стовпець k , якщо $S_{m+1k} = \max\{S_{m+1j}\} > 0$ (при пошуку мінімуму) або $S_{m+1k} = \min\{S_{m+1j}\} < 0$ (при пошуку максимуму);
- у якості розрішаючого рядка вибирається рядок r , що має мінімальне додатнє значення відношення $\alpha_r = S_{r0}/S_{rk}$.

Двоїсте завдання лінійного програмування формулюється наступним чином: завдання мінімізації цільової функції $L(\vec{x}) = c_1x_1 + \dots + c_nx_n$ або $L(\vec{x}) = \vec{c}^T \cdot \vec{x}$ при обмеженнях

$$\begin{cases} a_{11}x_1 + \dots + a_{1n}x_n \geq b_1 \\ \vdots \\ a_{m1}x_1 + \dots + a_{mn}x_n \geq b_m \end{cases} \quad \text{або} \quad A \cdot \vec{x} \geq \vec{b} \quad (2.1.14)$$

та завдання максимізації цільової функції $T(\vec{y}) = b_1y_1 + \dots + b_my_m$ або $T(\vec{y}) = \vec{b}^T \vec{y}$ при обмеженнях

$$\begin{cases} a_{11}y_1 + \dots + a_{m1}y_m \leq c_1 \\ \vdots \\ a_{1n}y_1 + \dots + a_{mn}y_m \leq c_n \end{cases} \quad \text{або} \quad A^T \cdot \vec{y} \leq \vec{c} \quad (2.1.15)$$

мають назви взаємодвоїстих завдань лінійного програмування. Зрозуміло, що у випадку коли завдання (2.1.15) розглядається як пряме, то відповідно завдання (2.1.14) є двоїстим до нього. Саме така постановка завдань розглядається у цій лабораторній роботі.

Якщо використовувати позначення симплекс-таблиці прямої задачі (2.1.10) що має розмір $(m+1) \times (n-m+1)$, то симплекс-таблиця двоїстої задачі виглядає як

$$\cdot \quad (2.1.16)$$

$$\beta = \max_i \min_j a_{ij}$$

і має розмір $(n-m+1) \times (m+1)$, тобто є транспонованою до вхідної зі зміною знаків деяких елементів.

Для взаємодвоїстих завдань необхідною і достатньою умовою оптимальності є умова $L(\vec{x}^*) = T(\vec{y}^*)$, при цьому $L_{\min} = T_{\max}$.

Постановка задачі

Дано

Проектується енергосистема, яка включає гідро та теплові станції (ГЕС та ТЕС). Задана цільова функція $T=c_1y_1+c_2y_2$, де c_1, c_2 – апіорно визначені коефіцієнти дольових вкладень сумарних потужностей y_1 та y_2 відповідно ГЕС та ТЕС у потужність генерації енергосистеми в цілому. Відомі питомі витрати ресурсів (водних, паливних, людських, капіталовкладень) на 10 МВт потужності $\{a_{ij}\}_{i=1..4;j=1..2}$ та ліміт $\{b_i\}_{i=1..4}$ за кожним ресурсом. Система обмежень задається у вигляді нерівностей $a_{i1}y_1+a_{i2}y_2 \leq b_i$, при цьому не можна перебільшувати певні обсяги відповідних ресурсів.

Потрібно

Прийняти оптимальне рішення: як розподіляти потужності ГЕС та ТЕС, котрі після побудування, забезпечують максимум заданої цільової функції T . Знайти значення оптимальних елементів рішення – потужностей ГЕС та ТЕС, а також величини відповідних цільових функцій, які отримані для прямої задачі графічним та симплекс методами та для двоїстої задачі симплекс методом

Рекомендації щодо виконання роботи

1. Описово сформулювати прикладне завдання лінійного програмування.
2. Навести математичне формулювання завдання прямої задачі лінійного програмування, при цьому систему обмежень представити як у вигляді системи нерівностей, так і системи рівнянь.
3. Знайти графічно оптимальне рішення прямої задачі.
4. Скласти вхідну симплекс-таблицю для рішення прямої задачі.
5. За допомогою наданого програмного забезпечення знайти опорне та оптимальне рішення прямої задачі. Зафіксувати зображення поточних виглядів симплекс-таблиць та вікна зі значеннями розрішуваних рядків та стовбців на кожному кроці максимізації цільової функції T .
6. Навести математичне формулювання завдання двоїстої задачі, при цьому систему обмежень представити як у вигляді системи нерівностей, так і системи рівнянь.
7. Скласти вхідну симплекс-таблицю для рішення двоїстої задачі.
8. За допомогою наданого програмного забезпечення знайти опорне та оптимальне рішення двоїстої задачі. Зафіксувати зображення поточних виглядів симплекс-таблиць та вікна зі значеннями розрішуваних рядків та стовбців на кожному кроці мінімізації цільової функції L .
9. Порівняти значення елементів рішення та цільових функцій, які отримані для прямої задачі графічним та симплекс методами та для двоїстої задачі симплекс методом.

Вимоги до звіту

Звіт по роботі повинен містити:

1. Назву дисципліни та лабораторної роботи.

2. Прізвище, ім'я та по батькові здобувача вищої освіти, код групи, номер варіанту завдання.
3. Описове формулювання прикладного завдання лінійного програмування.
4. Математичне подання прямої задачі лінійного програмування з представленнями обмежень як у вигляді системи нерівностей, так і системи рівнянь.
5. Графічне рішення прямої задачі, що повинне містити: графіки всіх лінійних обмежень, виділену область припустимих рішень, положення опорної прямої цільової функції і прямої цільової функції, яка перетинає точку оптимального рішення, точку (або множину точок) оптимального рішення, відкладені по осях графічного поля оптимальні значення елементів рішення – потужностей ГЕС та ТЕС.
6. Вхідну симплекс-таблицю для рішення прямої задачі.
7. Знімки зображень екрану, які виникають при знаходженні опорного та оптимального рішень прямої задачі, що включають поточні вигляди симплекс-таблиць та вікна зі значеннями розрішаючих рядків та стовбців на кожному кроці максимізації цільової функції.
8. Математичне подання двоїстої задачі з представленнями обмежень як у вигляді системи нерівностей, так і системи рівнянь.
9. Вхідну симплекс-таблицю для рішення двоїстої задачі.
10. Знімки зображень екрану, які виникають при знаходженні опорного та оптимального рішень двоїстої задачі, що включають поточні вигляди симплекс-таблиць та вікна зі значеннями розрішаючих рядків та стовбців на кожному кроці максимізації цільової функції.
11. Висновки, які містять порівняння значень елементів рішення та цільових функцій, які отримані для прямої задачі графічним та симплекс методами та для двоїстої задачі симплекс методом.

Контрольні питання і завдання

1. Дайте визначення рангу матриці. В яких межах може змінюватися ранг прямокутної матриці розміром $m \times n$, якщо принаймні один її елемент є ненульовим?
2. Дайте визначення сумісності та визначеності системи лінійних рівнянь та сформулюйте критерії їх виконання.
3. Приведіть приклади несумісної та невизначеної систем лінійних рівнянь. Знайдіть аналітичний розв'язок невизначеної системи.
4. Дайте визначення базисних і вільних змінних невизначеної систем лінійних рівнянь.
5. Математично сформулюйте основне завдання лінійного програмування у вигляді прямої та двоїстої задач.
6. Дайте визначення рішень завдання лінійного програмування: припустимого, опорного, оптимального.

7. Сформулюйте критерії вибору розрішаючого рядка та стовбця при знаходженні опорного та оптимального рішень при розв'язанні завдання лінійного програмування симплекс-методом. Поясніть чи є відмінними ці критерії для завдань мінімізації та максимізації при пошуку як опорного, так і оптимального рішень.
8. Сформулюйте критерії досягнення опорного та оптимального рішень при розв'язанні завдання лінійного програмування симплекс-методом. Поясніть чи є відмінними ці критерії для завдань мінімізації та максимізації при пошуку як опорного, так і оптимального рішень.

Варіанти завдань

Варіанти завдань мають літерно-цифровий код, наприклад А2. Літера означає варіант коефіцієнтів системи обмежень, а цифра – вид цільової функції.

Таблиця 2.1.1 Формат представлення коефіцієнтів обмежень завдання лінійного програмування

Вид ресурсу	Потреби ГЕС	Потреби ТЕС	Ліміт ресурсу
Водні ресурси, км ³	a_{11}	a_{12}	b_1
Органічне паливо, т.	a_{21}	a_{22}	b_2
Трудові ресурси, людей	a_{31}	a_{32}	b_3
Обсяг фінансування, млн. грн.	a_{41}	a_{42}	b_4

Таблиця 2.1.2 Коефіцієнти обмежень завдання лінійного програмування

А)	10		1		300	Б)	5		1		200	В)	4		1		500
	-		7		110		-		5		100		-		1		300
	1		3		210		1		5		100		1		1		150
	5		1		200		3		1		150		4		1		300
Г)	3		1		250	Д)	7		2		600	Е)	6		1		430
	-		2		400		-		5		300		-		3		250
	1		2		50		1		4		180		1		4		200
	2		1		400		1		1		160		6		1		600

Цільові функції:

- 1) $T=y_1+y_2$; 2) $T=3y_1+y_2$; 3) $T=2y_1+y_2$; 4) $T=4y_1+3y_2$.

2.2. Лабораторна робота № 2

Прийняття рішень при розв'язанні транспортних завдань

Об'єкт – транспортні завдання. Предмет – лінійні транспортні завдання за критерієм вартості. Мета – ознайомлення з методами прийняття рішень при розв'язанні транспортних завдань.

Стислі теоретичні відомості

Лінійне транспортне завдання (ТЗ) за критерієм вартості з правильним балансом формулюється наступним чином. Дано: $\{a_i\}_{i=1..m}$ – запаси товару у постачальників, $\{b_j\}_{j=1..n}$ – потреби товару споживачами, $\{c_{ij}\}_{i=1..m, j=1..n}$ – витрати по перевезенню одиниці товару від i -го постачальника до j -го споживача. При цьому виконуються обмеження

$$\sum_{i=1}^m x_{ij} = b_j, \quad \sum_{j=1}^n x_{ij} = a_i, \quad \sum_{i=1}^m a_i = \sum_{j=1}^n b_j, \quad (2.2.1)$$

де $\{x_{ij}\}_{i=1..m, j=1..n}$ – обсяги перевезень товару від i -го постачальника до j -го споживача. Потрібно скласти план перевезень: $\{x_{ij}\}_{i=1..m, j=1..n}$, $x_{ij} \geq 0$, який задовольняє вимогу

$$W = \sum_{j=1}^n \sum_{i=1}^m c_{ij} x_{ij} \rightarrow \min. \quad (2.2.2)$$

Це завдання лінійного програмування, в якому в силу умов (2.2.1), коефіцієнти при невідомих дорівнюють 1. число невідомих $m \cdot n$, число обмежуючих рівностей $m+n$, при цьому в силу третьої рівності в (2.2.1) тільки $m+n-1$ рівнянь лінійно незалежні. Таким чином серед невідомих $\{x_{ij}\}_{i=1..m, j=1..n}$ $m+n-1$ є базисними, а $n \cdot m - (m+n-1) = (m-1)(n-1)$ вільними. ТЗ як окремий випадок ОЗЛП можна вирішувати симплекс методом, проте розроблені методи, що дозволяють при невеликих значеннях n і m вирішувати ТЗ навіть вручну.

Знаходження опорного рішення. Для вирішення транспортного завдання використовується транспортна таблиця (табл. 2.2.1).

Таблиця 2.2.1. Загальний вигляд транспортної таблиці

b_j	b_1	$\dots$	b_n
a_i	c_{11}	$\dots$	c_{1n}
a_1	c_{11}	$\dots$	c_{1n}
$\vdots$	$\vdots$	$\dots$	$\vdots$
a_m	c_{m1}	$\dots$	c_{mn}

При побудові опорного рішення, як правило, використовується метод “північно-західного кута” або метод “мінімального елемента”.

У методі “північно-західного кута” побудови рішення починається з лівого верхнього кута таблиці – “північно-західного кута”. При цьому $x_{11}=\min\{a_1,b_1\}$. Якщо $a_1>b_1$ стовбець 1 прокреслюється, а замість a_1 записується залишок a_1-b_1 , після чого знаходиться значення $x_{12}=\min\{a_1-b_1,b_2\}$. Якщо $a_1<b_1$ рядок 1 прокреслюється, а замість b_1 записується залишок b_1-a_1 , після чого знаходиться значення $x_{21}=\min\{b_1-a_1,a_2\}$. Зазначені дії виконуються до заповнення таблиці, коли всі товари постачальників будуть розподілені по споживачам. При цьому переміщення по таблиці здійснюється праворуч-донизу (табл. 2.2.2).

Таблиця 2.2.2. Приклад побудови опорного рішення методом “північно-західного кута”

$b_j \backslash a_i$	60	70	110
150	60 6	70 10	20 4
90	12	2	70 8

Загальні витрати на перевезення при цьому складають

$$W = \sum_{j=1}^n \sum_{i=1}^m c_{ij} x_{ij} = 1860.$$

Метод “мінімального елемента” дозволяє отримати опорне рішення більш близьке до оптимального. Побудова рішення починається з клітини таблиці з мінімальним значенням c_{ij} . При цьому $x_{ij}=\min\{a_i,b_j\}$. Якщо $a_i>b_j$ стовбець i прокреслюється, а замість a_i записується залишок a_i-b_j , після чого переходять до комірки з наступним мінімальним значенням c_{ij} . Якщо $a_i<b_j$, прокреслюють стовбець j , замість b_j записується залишок b_j-a_i , після чого переходять до клітки з наступним найменшим значенням c_{ij} (табл. 2.2.3).

Таблиця 2.2.3. Приклад побудови опорного рішення методом “мінімального елемента”

$b_j \backslash a_i$	60	70	110
150	40 6	10	110 4
90	20 12	70 2	8

Загальні витрати на перевезення при цьому складають $W=1060$.

Опорні рішення побудовані. Для цього завдання $m=2$, $n=3$. Число базисних змінних $n+m-1 = 4$, число вільних $(m-1)(n-1) = 2$. Базисні змінні додатні, а вільні дорівнюють 0. Але можливо, що і базисні змінні будуть дорівнювати 0. Цей випадок виникає, коли залишок по рядку a_i і стовбцю b_j збігаються, $a_i = b_j$, і носить назву виродженого випадку. При цьому, згідно сформульованих правил, начеб то потрібно прокреслювати і рядок і стовбець, в якому знаходиться елемент рішення $x_{ij}=a_i=b_j$. Прокреслювання має на увазі, що всі елементи рядка або стовбця, значення яких ще не визначені, покладаються рівними 0 і залишаються вільними. Однак в виродженому випадку одне з не знайдених значень змінних в прокресленому рядку або стовбці слід залишити серед базисних і у відповідній комірці в явному вигляді записати 0. Для опорного рішення це не має значення, проте може бути суттєво при побудові оптимального рішення і дозволяє уникнути “патових” ситуацій, коли можливості оптимізації (які полягають в обміні базисних змінних на вільні) вичерпані, а оптимальне рішення ще не знайдено (табл. 2.2.4).

Таблиця 2.2.4. Приклад побудови опорного рішення методом “мінімального елемента”

b_j a_i	60	70	110	20
170	60 6	 10	110 4	 5
90	— 12	70 2	0 5	20 3

Кількість базисних і вільних змінних при цьому відповідно складає $n+m-1=5$, $(n-1)(m-1)=3$.

Випадки виродження транспортної матриці можуть мати місце і при побудові оптимального рішення.

Знаходження оптимального рішення. При знаходженні оптимального рішення найбільшого поширення набув метод, званий методом потенціалів. $\{a_i\}_{i=1..m}$ – запаси товару у постачальників, $\{b_j\}_{j=1..n}$ – потреби товару споживачами, При цьому кожному постачальнику i присвоюється певна величина $\{\alpha_i\}_{i=1..m}$, а споживачу j – $\{\beta_j\}_{j=1..n}$, які зветься потенціалами пунктів і які можна інтерпретувати як величини платежів за перевезення третій особі – перевізнику. Назвемо величину $z_{ij}=\alpha_i+\beta_j$ псевдовартістю перевезення між пунктами i та j . Покажемо, що для заданої сукупності платежів $\{\alpha_i\}_{i=1..m}$ та $\{\beta_j\}_{j=1..n}$ і при будь-якому припустимому плані перевезення величина повної псевдовартості не залежить від плану перевезень.

$$L = \sum_{j=1}^n \sum_{i=1}^m z_{ij} x_{ij} = \text{const.} \quad (2.2.3)$$

Дійсно, з урахуванням обмежень (2.2.1) можна записати

$$L = \sum_{j=1}^n \sum_{i=1}^m z_{ij} x_{ij} = \sum_{j=1}^n \sum_{i=1}^m (\alpha_i + \beta_j) x_{ij} = \sum_{j=1}^n \sum_{i=1}^m \alpha_i x_{ij} + \sum_{j=1}^n \sum_{i=1}^m \beta_j x_{ij} =$$

(2.2.4)

$$= \sum_{i=1}^m \alpha_i \sum_{j=1}^n x_{ij} + \sum_{j=1}^n \beta_j \sum_{i=1}^m x_{ij} = \sum_{i=1}^m \alpha_i a_i + \sum_{j=1}^n \beta_j b_j = \text{const.}$$

Тепер покажемо, що якщо для всіх базисних комірок плану $x_{ij} \geq 0$ $z_{ij} = c_{ij}$, а для всіх вільних $x_{ij} = 0$ $z_{ij} \leq c_{ij}$ або $d_{ij} = c_{ij} - z_{ij} \geq 0$, план є оптимальним. Вартість даного плану, виражена через базисні змінні, з урахуванням (2.2.4) складає

$$W = \sum_{j=1}^n \sum_{i=1}^m c_{ij} x_{ij} = \sum_{j=1}^n \sum_{i=1}^m z_{ij} x_{ij} = \text{const.} \quad (2.2.5)$$

З іншого боку, вартість цього ж плану через вільні змінні виражається як для будь-якого завдання лінійного програмування наступним чином

$$W = \sum_{j=1}^n \sum_{i=1}^m d_{ij} x_{ij} + \text{const.} \quad (2.2.6)$$

Оскільки для вільних змінних $x_{ij} = 0$ і $d_{ij} \geq 0$, неможливо поліпшити план за рахунок їх зміни, тобто він є оптимальним.

Таким чином, критерієм досягнення оптимальності плану перевезень є те, що для всіх базисних комірок плану виконуються умови $x_{ij} \geq 0$, $z_{ij} = c_{ij}$, а для всіх вільних $x_{ij} = 0$, $z_{ij} \leq c_{ij}$ або $d_{ij} = c_{ij} - z_{ij} \geq 0$.

Тепер розглянемо задачу визначення значень потенціалів пунктів. Число рівнянь виду $c_{ij} - z_{ij} = \alpha_i + \beta_j$ визначається числом базисних змінних і дорівнює $n+m-1$. Число потенціалів, які підлягають визначенню, дорівнює $n+m$. Один потенціал є вільною змінною і його значення можна вибрати довільно. При цьому, за звичай потенціал першого постачальника приймається таким, що дорівнює нулю. Приклад визначення потенціалів пунктів за допомогою транспортної таблиці приведений в табл. 2.2.5.

Таблиця 2.2.5. Приклад визначення потенціалів пунктів за допомогою транспортної таблиці

b_j	10	22	9	34	6	41	5	20
a_i								
0	31	18	10	7	13	6		8
-1	48		5	6	28	5	20	4
-2	38	4	8	34	7	6		7

Відповідна табл. 2.2.4 система рівнянь для визначення потенціалів пунктів має вигляд:

$$\alpha_1=0; \alpha_1+\beta_1=10; \beta_1=10; \alpha_1+\beta_3=6; \beta_3=6; \alpha_3+\beta_1=8; \alpha_3=-2; \\ \alpha_2+\beta_3=5; \alpha_2=-1; \alpha_3+\beta_2=7; \beta_2=9; \alpha_2+\beta_4=4; \beta_4=5.$$

Надалі дані в комірках, що відповідають маршрутам доцільно представляти у вигляді, що поданий на рис. 2.2.1.

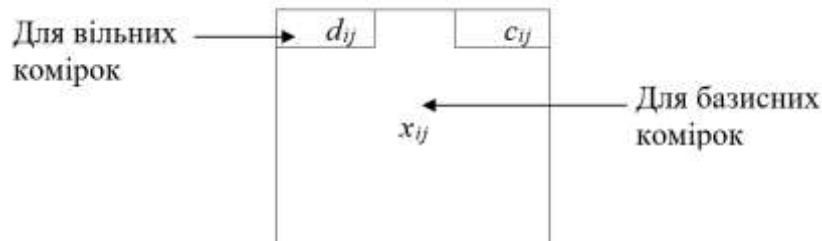


Рис. 2.2.1. Представлення даних у комірках транспортної таблиці

Подальший пошук оптимального рішення полягає в послідовній заміні базисних змінних до тих пір, поки не будуть виконані вимоги зазначеного вище критерія досягнення оптимальності плану перевезень. При цьому в базис вводиться вільна змінна, для якої виконується умова $d_{ij} < 0$. Заміна базисної змінної на вільну здійснюється шляхом циклічного перерахунку значень x_{ij} зі збереженням значень a_i і b_j по рядках і стовпцях, в яких проводиться перерахунок. При цьому обсяг перевізної сировини ξ перекидається з одного напрямку перевезення на інше. Цикл починається в вершині $d_{ij} < 0$, яка відповідає вільній змінній вводиться в базис. Обсяг товару на цьому напрямку збільшується (позитивна вершина) на ξ і стає рівним $x_{ij}=\xi$. Величина ξ визначається мінімальним значенням обсягів перевезень, що стоять в негативних вершинах кута. Змінна відповідає цьому мінімального значення переводиться з базисних в вільні. Зауважимо, що якщо мінімальні значення одночасно знаходяться в декількох негативних вершинах, виникає випадок виродження транспортної матриці. При цьому базисна змінна, що переводиться в вільні, вибирається довільно з змінних, що обнуляються. На рис. 2.2.2 наведено найпростіший цикл перерахунку з 4-ма вершинами.

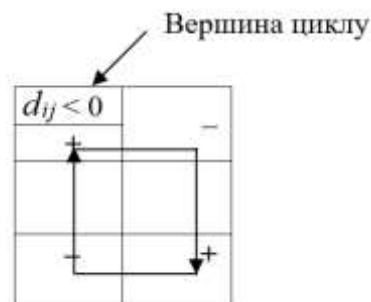


Рис. 2.2.2. Цикл перерахунку обсягів перевезень з 4-ма вершинами

Можливі також цикли перерахунку з 6-ма (рис. 2.2.3) і 8-ю вершинами (рис. 2.2.4).

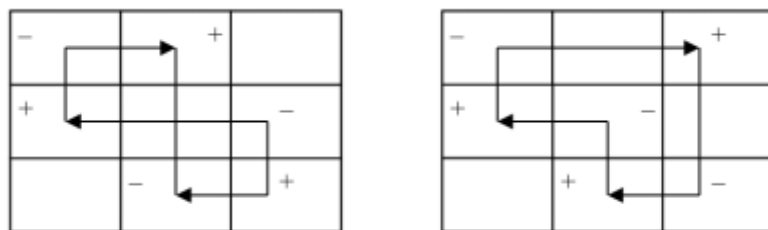


Рис. 2.2.3. Цикли перерахунку обсягів перевезень з 6-ма вершинами

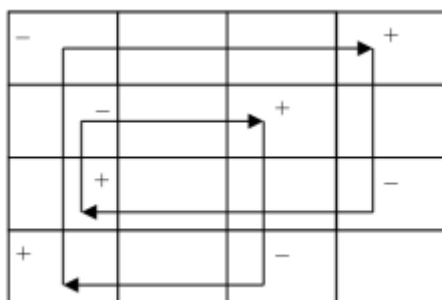


Рис. 2.2.4. Цикл перерахунку обсягів перевезень з 8-ю вершинами

В табл. 2.2.6 а, б приведений приклад реалізації циклу перерахунку обсягів перевезень в транспортній таблиці з вказівками обсягів перевезень відповідно до і після перерахунку.

Таблиця 2.2.6 а. Приклад реалізації циклу перерахунку обсягів перевезень в транспортній таблиці ($\xi=20$)

$b_j \backslash a_i$	60	70	110
150	40		110
90	20	70	

Таблиця 2.2.6 б. Приклад реалізації циклу перерахунку обсягів перевезень в транспортній таблиці

$b_j \backslash a_i$	60	70	110
150	60		90
90		70	20

ТЗ с неправильним балансом. В таких ТЗ порушується остання рівність умов (2.2.1). В завданнях з надлишком запасів сумарні обсяги запасів постачальників перевищують сумарні обсяги потреб споживачів. Вони зводяться до завдань з правильним балансом шляхом введення фіктивного споживача з нульовою вартістю перевезення в його адресу і обсягом заявки

$$b_{\phi} = \sum_{i=1}^m a_i - \sum_{j=1}^n b_j \text{ при } c_{i\phi} = 0. \quad (2.2.7)$$

При цьому фактично перевезення в адресу фіктивного споживача не здійснюються.

В завданнях з надлишком заявок сумарні обсяги потреб споживачів перевищують сумарні обсяги запасів постачальників. Вони зводяться до завдань з правильним балансом шляхом введення фіктивного постачальника з нульовою вартістю перевезення з його адреси і обсягом запасів

$$a_{\phi} = \sum_{j=1}^n b_j - \sum_{i=1}^m a_i \text{ при } c_{\phi j} = 0. \quad (2.2.8)$$

При цьому фактично перевезення з адреси фіктивного постачальника не здійснюються.

Іншим варіантом вирішення таких завдань є пропорційне зрізання всіх заявок, при цьому обсяги заявок, які будуть фактично задовольнятися розраховуються як

$$b_j' = b_j \frac{\sum_{i=1}^m a_i}{\sum_{j=1}^n b_j}. \quad (2.2.9)$$

Постановка задачі

Дано

Вхідна транспортна таблиця для лінійного завдання за критерієм вартості з правильним балансом.

Потрібно

Побудувати опорні та оптимальні рішення лінійних транспортних завдань за критерієм вартості з правильним та неправильним балансами.

Рекомендації щодо виконання роботи

1. Обрати вхідну транспортну таблицю згідно варіанту завдання і переконатися у правильності балансу її даних.
2. Вручну побудувати опорне рішення завдання, використовуючи метод, який визначається згідно варіанту завдання. Визначити кількість базисних та вільних комірок транспортної таблиці. Знайти: потенціали постачальників, споживачів та для вільних комірок транспортної таблиці різниці між вартостями та

псевдовартостями перевезень за відповідними маршрутами, а також повну вартість перевезень за отриманим планом. Зробити висновок відносно оптимальності отриманого опорного рішення.

3. За допомогою наданого програмного забезпечення знайти опорне та оптимальне рішення завдання з правильним балансом. Зафіксувати зображення поточних виглядів вікна програми з транспортною таблицею на кожному кроці оптимізації, включаючи такі її стани: вхідний, безпосередній після отримання опорного рішення, передуючий кожній наступній циклічній перестановці, після отримання оптимального рішення. Вручну знайти повну вартість перевезень для транспортної таблиці, яка відповідає опорному рішення.
4. Перевести вхідне транспортне завдання в завдання з надлишком запасів, вважаючи стовбець з номером $N \bmod 4$ таким, що відповідає фіктивному споживачеві. Тут N – номер здобувача вищої освіти за журналом групи. Слід мати на увазі, що в програмному забезпеченні, що надається, індексація рядків і стовбців транспортної таблиці починається з 0. Повторити дії п.3.
5. Перевести вхідне транспортне завдання в завдання з надлишком заявок, вважаючи рядок з номером $N \bmod 3$ таким, що відповідає фіктивному постачальникові. Тут N – номер здобувача вищої освіти за журналом групи. Повторити дії п.3.
6. Порівняти значення:
 - елементів рішення (вартості перевезень за відповідними маршрутами), потенціали постачальників, споживачів, різниці між вартостями та псевдовартостями перевезень, повні вартості перевезень для транспортних таблиць, які відповідають опорним рішенням і отримані за п.2, 3;
 - повні вартості перевезень для транспортних таблиць, які відповідають опорним рішенням і отримані за п. 2 – 5;
 - повні вартості перевезень для транспортних таблиць, які відповідають оптимальним рішенням і отримані за п. 2 – 5.

Вимоги до звіту

Звіт по роботі повинен містити:

1. Назву дисципліни та лабораторної роботи.
2. Прізвище, ім'я та по батькові здобувача вищої освіти, код групи, номер варіанту завдання.
3. Транспортну таблицю, яка відповідає опорному рішенням і отримана за п. 2 рекомендацій щодо виконання роботи, а також повну вартість перевезень за отриманим планом. Систему рівнянь для отримання потенціалів постачальників та споживачів. Висновок відносно оптимальності отриманого опорного рішення.
4. Знімки зображень екрану, які виникають при знаходженні опорних та оптимальних рішень за п. 3 – 5. рекомендацій щодо виконання

роботи, а також повні вартості перевезень за отриманими опорними планами.

5. Висновки, які містять порівняння результатів за п 6 рекомендацій щодо виконання роботи.

Контрольні питання і завдання

1. Приведіть математичну постановку лінійного транспортного завдання за критерієм вартості з правильним балансом. Поясніть сенс кожного з обмежень завдання. Наведіть формули для кількості змінних елементів рішення, кількості базисних та вільних змінних.
2. Приведіть структуру транспортної таблиці для завдання за критерієм вартості. Вкажіть дані, які вона містить з початку та які виникають при побудуванні опорного та оптимального рішень.
3. Сформулюйте принципи побудови опорного рішення транспортного завдання методами “північно-західного кута” та “мінімального елемента”.
4. Поясніть у якому разі виникають випадки виродження транспортної таблиці в процесі оптимізації, до яких утруднень вони можуть призвести і як потрібно коректно обробляти такі випадки.
5. Поясніть у чому полягає метод потенціалів, який використовується при побудуванні оптимального рішення транспортного завдання за критерієм вартості та який сенс мають потенціали постачальників, споживачів та псевдовартість перевезень.
6. Сформулюйте критерій досягнення оптимальності плану перевезень транспортного завдання за критерієм вартості.
7. Поясніть яким чином здійснюються кроки оптимізації при побудуванні оптимального рішення транспортного завдання за критерієм вартості
8. Поясніть які існують види транспортних завдань з неправильним балансом і які способи їх розв’язання.

Варіанти завдань

Варіанти завдань мають літерно-цифровий код, наприклад А2. Літера означає метод побудови опорного рішення (А – “північно-західного кута” та Б – “мінімального елемента”), а цифра – дані вхідної транспортної таблиці (табл. 2.2.7).

Таблиця 2.2.7. Вхідна транспортна таблиця

1)	30	65	25	10
50	5	13	6	11
35	9	2	3	10
45	11	6	13	5

2)	30	60	15	25
20	5	11	2	7
40	12	3	6	8

70	4	9	6	10
----	---	---	---	----

3)	20	20	45	45
35	9	4	6	10
65	12	6	3	8
30	11	2	5	7

4)	60	30	20	30
30	5	13	6	11
45	9	2	3	10
65	4	7	12	8

5)	15	60	25	30
50	9	2	3	10
45	4	7	12	8
35	11	6	13	5

6)	30	65	25	10
50	9	2	3	10
35	4	7	12	8
45	11	6	13	5

7)	30	60	15	25
20	5	11	2	7
40	9	4	6	10
70	4	7	12	8

8)	20	20	45	45
35	9	4	6	10
65	12	6	3	8
30	11	2	5	7

9)	60	30	20	30
30	5	11	2	7
45	12	3	6	8
65	4	9	6	10

10)	15	60	25	30
50	5	13	6	11
45	9	2	3	10
35	4	7	12	8

2.3. Лабораторна робота № 3

Розв'язання завдань нелінійного програмування графічним методом

Об'єкт – методи розв'язання завдань нелінійного програмування.

Предмет – графічний метод розв'язання завдань нелінійного програмування.

Мета – отримання навичок використання графічного метода розв'язання завдань нелінійного програмування з двома вільними змінними.

Стислі теоретичні відомості

Необхідною умовою екстремуму функції f довільної кількості незалежних змінних n в точці $\vec{x}^*$ є рівність до нуля всіх її часткових похідних

$$\left. \frac{\partial f}{\partial x_i} \right|_{\vec{x}=\vec{x}^*} = 0 \quad i = 1..n \quad (2.3.1)$$

Область G простору довільної розмірності є опуклою якщо для будь яких двох крапок, які належать до неї виконується умова

$$k \cdot \vec{x}_1 + (1-k) \cdot \vec{x}_2 \in G \quad \forall \vec{x}_1, \vec{x}_2 \in G, \quad 0 < k < 1. \quad (2.3.2)$$

Нехай G є опуклою множиною простору довільної розмірності. Функція f , яка визначена на множині G зветься опуклою, якщо для будь яких двох крапок, які належать множині G виконується умова

$$f[k \cdot \vec{x}_1 + (1-k) \cdot \vec{x}_2] \leq k \cdot f(\vec{x}_1) + (1-k) \cdot f(\vec{x}_2) \quad \forall \vec{x}_1, \vec{x}_2 \in G, 0 < k < 1. \quad (2.3.3)$$

Функція f , яка визначена на множині G зветься увігнутою (опуклою вгору), якщо для будь яких двох крапок, які належать G виконується умова

$$f[k \cdot \vec{x}_1 + (1-k) \cdot \vec{x}_2] \geq k \cdot f(\vec{x}_1) + (1-k) \cdot f(\vec{x}_2) \quad \forall \vec{x}_1, \vec{x}_2 \in G, 0 < k < 1. \quad (2.3.4)$$

Якщо нерівності (2.3.3), (2.3.4) є суворими то функція f є відповідно суворо увігнутою і суворо опуклою.

Розглянемо множину визначників

$$\Delta_1 = a_{11}, \quad \Delta_2 = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}, \quad \Delta_3 = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}, \dots \Delta_n = \begin{vmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \vdots & \vdots \\ a_{n1} & \dots & a_{nn} \end{vmatrix} \quad (2.3.5)$$

$$a_{ij} = \left. \frac{\partial^2 f}{\partial x_i \partial x_j} \right|_{\vec{x}=\vec{x}^*} \quad i, j = 1..n.$$

Умовою суворої опуклості функції f в околиці точки $\vec{x}^*$ є додатність всіх визначників (2.3.5), умовою суворої увігнутості є від'ємність визначника і чергування знаків значень наступних визначників.

Умови суворої опуклості (суворої увігнутості) є достатніми умовами мінімуму (максимуму) функції f в точці $\vec{x}^*$.

Для функції двох незалежних змінних $f=f(x_1, x_2)$, оскільки значення змішаної похідної не залежить від порядку диференціювання, достатня умова екстремуму виглядає як

$$\Delta_2 = a_{11}a_{22} - a_{12}^2 > 0. \quad (2.3.6)$$

При цьому у випадку $a_{11}>0$ має місце мінімум, при $a_{11}<0$ – максимум. У разі $\Delta_2<0$ екстремуму немає, а якщо $\Delta_2=0$ потрібне додаткове дослідження

Постановка задачі

Дано

Система обмежень та цільова функція.

Потрібно

Знайти графічним методом значення елементів рішення, які забезпечують мінімізацію та максимізацію цільової функції та її значення в мінімумі та максимумі.

Рекомендації щодо виконання роботи

1. Побудувати графічне поле, яке містить всі графіки функцій обмежень та ізолінії цільової функції, що відповідають значенням мінімуму та максимуму. Побудова може здійснюватися як з використанням середовища математичних розрахунків так і вручну. Визначити значення елементів рішення і цільової функції в точках мінімуму та максимуму.
2. Розрахувати значення елементів рішення і цільової функції в точках мінімуму та максимуму за допомогою вбудованих функцій оптимізації середовища математичних розрахунків.

Вимоги до звіту

Звіт по роботі повинен містити:

1. Назву дисципліни та лабораторної роботи.
2. Прізвище, ім'я та по батькові здобувача вищої освіти, код групи, номер варіанту завдання.
3. Постановку завдання нелінійного програмування.
4. Графічне поле, яке містить всі графіки функцій обмежень та ізолінії цільової функції, що відповідають значенням мінімуму та максимуму.
5. Код програми, що реалізує поставлене завдання за допомогою вбудованих функцій оптимізації середовища математичних розрахунків.
6. Висновки, які містять порівняння результатів розв'язання задачі нелінійного програмування графічним методом та за допомогою

вбудованих функцій оптимізації середовища математичних розрахунків.

Контрольні питання і завдання

1. Дайте визначення опуклої області простору довільної розмірності.
2. Дайте визначення опуклої та увігнутої функцій.
3. Сформулюйте необхідну та достатню умови екстремуму функцій багатьох змінних.
4. Для функцій $z(x,y)=x^2+y^2$, $z(x,y)=x \cdot y$, $z(x,y)=x^2+y^2$ дослідити наявність екстремуму в точці $(x,y)=(0,0)$.
5. Для функції $z(x,y)=x^2-10x+y^2=x^2-2 \cdot 5x+25+y^2-25=(x-5)^2+y^2-25$ побудувати сімейство ізоліній $z(x,y)=\text{const}$, для яких значення const становить: -25, -21, -16, -9, 0, 11.

Варіанти завдань

Таблиця 2.3.1. Система обмежень та цільова функція

№	Система обмежень	Цільова функція
1	$\begin{cases} 2x+3y \leq 30 \\ x+2y \leq 15 \\ x \geq 0 \quad y \geq 0 \end{cases}$	$y - x^2 + 4x$
2	$\begin{cases} x^2 + 2x + y^2 - 2y - 14 \geq 0 \\ 2x + y \leq 10 \\ x \geq 0 \quad y \geq 0 \end{cases}$	$x \cdot y$
3	$\begin{cases} x^2 - 2x + y^2 - 2y - 34 \geq 0 \\ x \geq 1 \quad y \geq 1 \end{cases}$	$4x + 3y$
4	$\begin{cases} x^2 + 2x + y^2 - 2y - 14 \geq 0 \\ 2x + y \leq 10 \\ x \geq 0 \quad y \geq 0 \end{cases}$	$9(x-5)^2 + 4(y-6)^2$
5	$\begin{cases} 6x + 4y \geq 12 \\ 2x + 3y \leq 24 \\ -3x + 4y \leq 12 \\ x \geq 0 \quad y \geq 0 \end{cases}$	$x \cdot y$
6	$\begin{cases} y \geq (x-4)^2 + 2 \\ -3x + 5y \leq 15 \\ x + 4y \geq 16 \\ x \geq 0 \quad y \geq 0 \end{cases}$	$x^2 - x + y^2$

7	$\begin{cases} 3x+4y \leq 25 \\ x \cdot y \geq 4 \\ x \geq 0 \ y \geq 0 \end{cases}$	$9(x-5)^2 + 4(y-6)^2$
8	$\begin{cases} 2x+4y \leq 16 \\ -4x+2y \leq 8 \\ x+3y \geq 9 \\ x \geq 0 \ y \geq 0 \end{cases}$	$x \cdot y$
9	$\begin{cases} 2x+y \leq 10 \\ -4x+3y \leq 6 \\ 2x+4y \geq 8 \\ x \geq 0 \ y \geq 0 \end{cases}$	$(x-2)^2 + (y-2)^2$
10	$\begin{cases} 3x+2y \geq 7 \\ 8x-y \leq 8 \\ -18x+4y \leq 12 \\ x \geq 0 \ y \geq 0 \end{cases}$	$(x-4)^2 + (y-3)^2$
11	$\begin{cases} x^2+(y-3)^2 \leq 9 \\ (x-3)^2 + y \leq 6 \\ x \geq 0 \ y \geq 0 \end{cases}$	$2x-6y$
12	$\begin{cases} x^2+(y-3)^2 \leq 9 \\ y \geq (x-4)^2 + 2 \\ y \leq -(x-4)^2 + 6 \\ x \geq 0 \ y \geq 0 \end{cases}$	$-x-y$
13	$\begin{cases} x^2+y^2 \leq 36 \\ y \geq (x-4)^2 + 2 \\ x \geq 0 \ y \geq 0 \end{cases}$	$x+y$
14	$\begin{cases} x \cdot y \geq 1 \\ y \leq -(x-4)^2 + 6 \\ x \geq 0 \ y \geq 0 \end{cases}$	$2x+y$

15	$\begin{cases} x^2 + (y-4)^2 \leq 16 \\ (x-4)^2 + y^2 \leq 16 \\ x \geq 0 \quad y \geq 0 \end{cases}$	$x + y$
16	$\begin{cases} x \geq 1 \\ x \leq y \end{cases}$	$x^2 + y^2$
17	$\begin{cases} x^2 + (y-1)^2 \leq 1 \\ x^2 + y^2 \leq 1 \end{cases}$	$x + y$
18	$\begin{cases} y \geq 2x^2 + 6 \\ -x + y \leq 8 \end{cases}$	$x^2 + y^2$
19	$\begin{cases} x^2 + y^2 \leq 25 \\ x \geq 0 \quad y \geq 0 \end{cases}$	$x \cdot y$
20	$\begin{cases} 4x^2 + y^2 \leq 16 \\ x \geq 0 \quad y \geq 0 \end{cases}$	$x^2 + y^2 - 6x - 8y$
21	$\begin{cases} x + y \leq 4 \\ x \geq 0 \quad y \geq 0 \end{cases}$	$(x-3)^2 + (y-2)^2$
22	$\begin{cases} (x-3)^2 + y^2 \leq 1 \\ x \geq 0 \quad y \geq 0 \end{cases}$	$x^2 - 2x \cdot y + y^2$
23	$\begin{cases} (x-2)^2 + y^2 \leq 1 \\ x^2 + (y-2)^2 \leq 4 \end{cases}$	$x + y$
24	$\begin{cases} x + 2y \leq 6 \\ x \cdot y \leq 0,5 \\ x \geq 0 \quad y \geq 0 \end{cases}$	$(x-1)^2 + (y-1)^2$
25	$\begin{cases} y \geq (x-2)^2 + 1 \\ y \leq x \end{cases}$	$x + y$
26	$\begin{cases} x + y \leq 6 \\ x \cdot y \geq 2 \\ x \geq 0 \quad y \geq 0 \end{cases}$	$x + 2y$
27	$\begin{cases} (x-2)^2 + 2y^2 \leq 4 \\ y \leq x-1 \\ x \geq 0 \quad y \geq 0 \end{cases}$	$x^2 + y^2 - 4x - 8y + 20$

28	$\begin{cases} y \geq (x-2)^2 \\ y \leq 4-(x-2)^2 \\ y \geq x \end{cases}$	$x + y$
----	--	---------

2.4. Лабораторна робота № 4

Розв'язання завдань нелінійного програмування методом Лагранжа

Об'єкт – методи розв'язання завдань нелінійного програмування.

Предмет – метод Лагранжа розв'язання завдань нелінійного програмування.

Мета – отримання навичок використання метода Лагранжа для розв'язання завдань нелінійного програмування.

Стислі теоретичні відомості

Метод Лагранжа розв'язання завдань нелінійного програмування відноситься до аналітичних. Завдання, які можуть вирішуватися за його допомогою, формулюються наступним чином: знайти невід'ємні значення змінних x_j ($j=1..n$) – елементів рішення, які при наявності обмежень у вигляді рівностей $q_i(x_1, \dots, x_n)=0$ ($i=1..m$) мінімізують, або максимізують цільову функцію $W=W(x_1, \dots, x_n)$.

Для розв'язання завдання складається функція Лагранжа

$$F(x_1, \dots, x_n, \lambda_1, \dots, \lambda_m) = W(x_1, \dots, x_n) + \sum_{i=1}^m \lambda_i q_i(x_1, \dots, x_n), \quad (2.4.1)$$

або

$$F(\vec{x}, \vec{\lambda}) = W(\vec{x}) + \sum_{i=1}^m \lambda_i q_i(\vec{x}), \quad (2.4.2)$$

де λ_i ($i=1..m$) – невизначені множники Лагранжа.

Можливі оптимальні значення елементів рішення знаходяться із системи рівнянь, яка випливає з необхідної умови екстремуму функції багатьох змінних (2.3.1)

$$\begin{cases} \frac{\partial F}{\partial x_j} = \frac{\partial W}{\partial x_j} + \sum_{i=1}^m \lambda_i \frac{\partial q_i}{\partial x_j} & i = 1..n, \\ \frac{\partial F}{\partial \lambda_i} = q_i = 0 & j = 1..m. \end{cases} \quad (2.4.4)$$

Це система з $n+m$ рівнянь з $n+m$ невідомими. В результаті її розв'язання знаходяться значення елементів рішення $\vec{x}^*$ в яких може досягатися екстремуму функції F . Оскільки рішення знаходяться при умовах $q_i=0$, при значеннях елементів рішення $\vec{x}^*$ може досягатися екстремуму і функції W . Зазначимо, що оскільки метод Лагранжа ґрунтується на необхідній, а не достатній умові екстремуму, власно екстремуму функції W у точці $\vec{x}^*$ може і не бути, тобто вона буде не точкою мінімуму або максимуму, а сідловою точкою функції W .

Постановка задачі

Дано

Система обмежень та цільова функція.

Потрібно

Знайти методом Лагранжа значення елементів рішення в точці, в якій може забезпечуватися мінімізація або максимізація цільової функції та побудувати графік її залежності в околиці цієї точки.

Рекомендації щодо виконання роботи

1. Записати функцію Лагранжа і відповідну систему рівнянь для розв'язання оптимізаційного завдання.
2. Розрахувати значення елементів рішення в точці, в якій може забезпечуватися мінімізація або максимізація цільової функції шляхом розв'язання системи рівнянь Лагранжа. Обчислення за цим пунктом і далі можуть здійснюватися як з використанням середовища математичних розрахунків так і вручну.
3. Вважаючи у системі рівнянь обмежень змінні x та y базисними, а z – вільною:
 - задати кілька десятків значень змінної z з постійним кроком в околиці значення, обчисленого за п.2;
 - записати рівняння обмежень таким чином, щоб базисні змінні знаходилися в лівих частинах, а вільна змінна і постійні члени – у правих;
 - розв'язати систему лінійних рівнянь обмежень відносно змінних x та y для всіх заданих вище значень змінної z , при цьому доцільно використовувати метод Крамера;
 - для всіх заданих вище значень змінної z підставити їх та отримані відповідні їм значення змінних x та y у ліві частини рівнянь обмежень, які записані у первісному вигляді і переконатися що всі вони приймають нульові значення.
4. Побудувати графіки залежностей цільової функції, змінних x , y та z від номеру кроку змінної z .
5. Зробити висновки про наявність екстремуму цільової функції для значення змінної z , обчисленого за п.2 та відповідності графікам змінних їх значенням, обчислених за п.2.

Вимоги до звіту

Звіт по роботі повинен містити:

1. Назву дисципліни та лабораторної роботи.
2. Прізвище, ім'я та по батькові здобувача вищої освіти, код групи, номер варіанту завдання.
3. Постановку завдання нелінійного програмування.
4. Систему рівнянь Лагранжа і її розв'язання.
5. Таблиці значень змінних x, y, z та лівих частин рівнянь обмежень за п.3 рекомендацій щодо виконання роботи.
6. Код програми, що реалізує п. 4 – 5 (при програмному розв'язанні завдання).
7. Графіки залежностей цільової функції, змінних x, y та z від номеру кроку змінної z .
8. Висновки про наявність екстремуму цільової функції для значення змінної z , обчисленого за п.2 та відповідності графікам змінних їх значенням, обчислених за п.2.

Контрольні питання і завдання

1. При яких умовах можна використовувати метод Лагранжа?
2. У загальному вигляді викладіть метод Лагранжа?
3. Чи гарантує метод Лагранжа отримання оптимального рішення? Відповідь обґрунтуйте.
4. Розв'яжіть методом Лагранжа завдання оптимізації цільової функції $W(x,y)=x^2+y^2$ при обмеженні $x+2y=4$. Наведіть ескіз графічного рішення.
5. Розв'яжіть методом Лагранжа завдання оптимізації цільової функції $W(x,y)=x^2-2x-y$ при обмеженні $x+y=2$. Наведіть ескіз графічного рішення.
6. Розв'яжіть методом Лагранжа завдання оптимізації цільової функції $W(x,y)=1/x+y$ при обмеженні $y-x=1$. Наведіть ескіз графічного рішення.
7. Визначте радіус та висоту бака циліндричної форми, виходячи з його об'єму V і мінімізації площі бокової поверхні.
8. Запишіть систему рівнянь Лагранжа для визначення радіуса та висоти бункера конічної форми з відкритим верхом, виходячи з його об'єму V і мінімізації площі бокової поверхні.

Варіанти завдань

Таблиця 2.3.1. Система обмежень та цільова функція

№	Система обмежень	Цільова функція
1	$\begin{cases} 2x+3y+z=12 \\ 3y-z=3 \end{cases}$	$5x^2 + x \cdot z + 3y^2 - 3y \cdot z + z^2$
2	$\begin{cases} 3x+z=14 \\ 2y+4z=7 \end{cases}$	$6x^2 + x + x \cdot z + 3y^2 - z^2 - y \cdot z$

3	$\begin{cases} x - z = -3 \\ 2x + y - 4z = 5 \end{cases}$	$x^2 - x \cdot y + 2y^2 + y \cdot z + 2z^2$
4	$\begin{cases} 4x + y - 3z = 9 \\ 2x + 5z = 16 \end{cases}$	$5x^2 + 2x \cdot y + y^2 + 4z^2 - z$
5	$\begin{cases} x + z = 6 \\ 2x - 4y - z = 2 \end{cases}$	$x^2 + 2x \cdot y + 3y^2 + y \cdot z + 2z^2$
6	$\begin{cases} x + 2y + z = 6 \\ 3x - 2y - z = -4 \end{cases}$	$2x^2 - 2x \cdot y + y^2 - 3y \cdot z - 4z^2$
7	$\begin{cases} 5x + y - 4z = 18 \\ 3x + 5z = 25 \end{cases}$	$3x^2 + 4x \cdot y - x \cdot z + y^2 + z^2$
8	$\begin{cases} 3x + y + 5z = 30 \\ 2x - y - 3z = 0 \end{cases}$	$x^2 + x \cdot y - 4x \cdot z - y^2 + z^2$
9	$\begin{cases} x - y - z = 6 \\ 3x + 5z = 8 \end{cases}$	$3x^2 + x - x \cdot z - 2y^2 + y \cdot z + 2z^2$
10	$\begin{cases} 3x + y - 5z = 8 \\ 4x - y + z = 3 \end{cases}$	$2x^2 - 2x \cdot y - x \cdot z + y^2 + 5z^2$
11	$\begin{cases} 4x - y - 3z = 12 \\ 5y + z = 8 \end{cases}$	$x^2 + x \cdot y + 3y^2 - y \cdot z + 4z^2$
12	$\begin{cases} 3x + y - 5z = -12 \\ 4x + 2y + z = 18 \end{cases}$	$2x^2 + x \cdot z - 43y^2 + y \cdot z + z^2$
13	$\begin{cases} 3x + 2y - 3z = 10 \\ 4y + 5z = 3 \end{cases}$	$x^2 + 2x + 5x \cdot y + y^2 - y \cdot z - 4z^2$
14	$\begin{cases} x - y + 5z = 16 \\ 4x + y - 4z = 20 \end{cases}$	$3x^2 + x \cdot y + x \cdot z + 2y^2 - 3z^2$
15	$\begin{cases} 2x - 3y - z = -8 \\ 4y + 5z = 18 \end{cases}$	$6x^2 + 5x \cdot z + 3y^2 + y \cdot z - 4z^2$
16	$\begin{cases} 4x + y - 2z = 15 \\ 4y + z = 8 \end{cases}$	$3x^2 - x \cdot y + y^2 - 3y \cdot z + z^2$
17	$\begin{cases} 5x + y - z = 8 \\ 4x - y + 3z = 16 \end{cases}$	$2x^2 + x \cdot z + y^2 - y \cdot z - 3z^2$
18	$\begin{cases} 2x + y - 4z = -6 \\ x - y + 5z = 8 \end{cases}$	$5x^2 + x \cdot z + 3y^2 - 3y \cdot z + z^2$
19	$\begin{cases} x - 4y + z = 12 \\ 4x + 3y - z = -6 \end{cases}$	$4x^2 + x \cdot z + x + 5y^2 + y \cdot z - 4z^2$

20	$\begin{cases} 3x + y - 4z = 16 \\ 2x + y + z = 22 \end{cases}$	$x^2 + x \cdot y + 3y^2 - y \cdot z - 4z^2$
21	$\begin{cases} 6x + 4y - z = -8 \\ 3y + z = 12 \end{cases}$	$2x^2 + x \cdot y + y^2 - 3y \cdot z - z^2$
22	$\begin{cases} 6x - y + 2z = 8 \\ 4x + y - z = -2 \end{cases}$	$3x^2 + x \cdot y + x \cdot z + y^2 - 2z^2$
23	$\begin{cases} 5x + y - 4z = 8 \\ 4x + 3y - 5z = 16 \end{cases}$	$x^2 - x \cdot y - x \cdot z + y^2 + y \cdot z - 3z^2$
24	$\begin{cases} x + y + 3z = 5 \\ 6x - y - 2z = 8 \end{cases}$	$6x^2 + 2x \cdot y - 3y^2 + y \cdot z + 4z^2$
25	$\begin{cases} 2x + y - 5z = -10 \\ x - y + 4z = 7 \end{cases}$	$x^2 - x \cdot z + 2y^2 + y \cdot z - 3z^2$
26	$\begin{cases} 2x + y - 4z = 16 \\ 5x - 4y + z = 9 \end{cases}$	$3x^2 + 5x + x \cdot y - x \cdot z + y^2 - 4z^2$
27	$\begin{cases} x - y + 3z = 8 \\ 4x + y = 5 \end{cases}$	$2x^2 + 5x \cdot z - y^2 + 4y + y \cdot z + 3z^2$
28	$\begin{cases} 2x + y - z = 12 \\ x - y + 5z = 8 \end{cases}$	$6x^2 + x \cdot y - 3x \cdot z + 2y^2 - 4z^2$
29	$\begin{cases} x - y - z = 8 \\ 5x + y - 2z = 0 \end{cases}$	$x^2 - 4x \cdot z + 2y^2 + 2y \cdot z + 3z^2$

2.5. Лабораторна робота № 5

Прийняття рішень в умовах невизначеності

Об'єкт – завдання прийняття рішень в умовах невизначеності. Предмет – завдання теорії статистичних рішень і парних ігор. Мета – ознайомлення з можливостями прийняття рішень за допомогою методів теорії статистичних рішень і парних ігор.

Стислі теоретичні відомості

При вирішенні ряду задач доводиться мати справу з невизначеністю, яка обумовлена імовірнісним характером факторів, що впливають на результат. Завдання такого роду досліджуються в теорії статистичних рішень. В інших завданнях аналізуються конфліктні ситуації, в яких стикаються інтереси двох або більше сторін, що переслідують різні інтереси, причому результат операції залежить від того, який спосіб дії або стратегію вибере кожен з противників. Такі завдання аналізуються в теорії конфліктних ситуацій або теорії

антагоністичних ігор. Завдання обох типів мають багато спільного. У певному сенсі завдання першого типу можна розглядати як гру зі стихією.

Гра – змодельована операція, в якій має місце фактор невизначеності. Як будь-яка модель, гра має певні абстракції. У грі може брати участь дві або більше сторони. У першому випадку гра називається парною, в другому – множинною. Учасники множинної гри можуть організовувати постійні або тимчасові коаліції.

Гра складається з елементарних дій, що здійснюються сторонами, які носять назву ходів. Розрізняють особисті ходи, які є результатом усвідомленого вибору боку, і випадкові ходи, які є випадковими подіями і можуть бути описані за допомогою імовірнісних розподілів. Ігри в шашки або шахи складається тільки з особистих ходів, гра в кістки – тільки з випадкових, картіжні ігри в “дурня” або в преферанс – як з особистих, так і з випадкових ходів. У парних іграх зі стихією остання робить тільки випадкові ходи. Гра підпорядковується правилам, які регламентують:

- можливі варіанти дій (стратегії гравців);
- обсяг інформації кожної сторони про поведінку іншої;
- результати гри, до яких призводить кожна сукупність ходів.

Наслідкам приписують хоча б умовну кількісну оцінку.

Якщо у кожного гравця є тільки кінцеве число стратегій, то гра називається кінцевою, в іншому випадку – нескінченною.

Матриця, в якій рядки відповідають стратегії першого гравця, стовпці – стратегії другого гравця, а елементами матриці є кількісні оцінки результатів, що відповідають обраної парі стратегій гравців, називається платіжною матрицею або матрицею гри. Крім чистих стратегій, які фігурують в платіжній матриці, розрізняють також змішані стратегії, що представляють собою випадкове чергування чистих стратегій.

Елементи теорії статистичних рішень. Розглянемо платіжну матрицю розміром $m \times n$, гри зі стихією.

Рядки матриці $\{A_i\}_{i=1,\dots,m}$ – стратегії гравця,

Стовпці матриці $\{B_j\}_{j=1,\dots,n}$ – випадкові варіанти обстановки, які можна інтерпретувати як стратегії стихії.

Елементи матриці $\{a_{ij}\}_{i=1,\dots,m;j=1,\dots,n}$ – виграші гравця. У деяких випадках в якості елементів платіжної матриці зручно вибирати витрати гравця $\{b_{ij}\}_{i=1,\dots,m;j=1,\dots,n}$.

Рішенням завдання буде стратегія максимізує виграш або мінімізує витрати.

Іноді при виборі стратегії більш інформативною, ніж платіжна матриця є матриця ризику. У разі, якщо елементами платіжної матриці є виграші, ризик являє собою величину $r_{ij} = \beta_j - a_{ij} = \max a_{ij} - a_{ij}$, тобто під ризиком при виборі стратегії i розуміється максимальний недоотриманий виграш у випадковому варіанті обстановки $\beta_j = \max a_{ij}$. Ризик є величиною невід'ємною.

Таблиця 2.5.1. Приклад платіжної матриці у вигляді матриці виграшів і відповідної матриці ризику

$A_i \backslash B_j$	B_1	B_2	B_3	B_4
A_1	1	4	5	9
A_2	3	8	4	3
A_3	4	6	6	2

$A_i \backslash B_j$	B_1	B_2	B_3	B_4
A_1	3	4	1	0
A_2	1	0	2	6
A_3	0	2	0	7

Коли ж елементами платіжної матриці є витрати, ризик являє собою величину $r_{ij} = b_{ij} - \beta_j = b_{ij} - \min b_{ij}$, тобто під ризиком при виборі стратегії i розуміються максимальні перевищені витрати у випадковому варіанті обстановки $\beta_j = \min b_{ij}$.

Таблиця 2.5.2. Приклад платіжної матриці у вигляді матриці витрат і відповідної матриці ризику

$A_i \backslash B_j$	B_1	B_2	B_3	B_4
A_1	1	4	5	9
A_2	3	8	4	3
A_3	4	6	6	2

$A_i \backslash B_j$	B_1	B_2	B_3	B_4
A_1	0	0	1	7
A_2	2	4	0	1
A_3	3	2	2	0

Критерії прийняття рішень, засновані на відомих імовірнісних умовах обстановки. Задані наступні величини $Q_j = P(B_j)$, $j = 1, \dots, n$ – ймовірності стратегії стихії.

$$\sum_{j=1}^n Q_j = 1. \quad (2.5.1)$$

В цьому випадку показником ефективності, який потрібно максимізувати, є математичне очікування виграшу при обраної стратегії:

$$\bar{a}_i = \sum_{j=1}^n Q_j a_{ij} \Rightarrow \max. \quad (2.5.2)$$

Як показники ефективності можна також вибрати математичне очікування витрат і ризику, які необхідно мінімізувати:

$$\bar{b}_i = \sum_{j=1}^n Q_j b_{ij} \Rightarrow \min, \quad \bar{r}_i = \sum_{j=1}^n Q_j r_{ij} \Rightarrow \min. \quad (2.5.3)$$

Покажемо, що стратегія, що максимізує середній виграш, мінімізує середній ризик: $\bar{a}_i + \bar{r}_i = \sum_{j=1}^n Q_j (a_{ij} + \beta_j - a_{ij}) = \text{const} \quad \bar{r}_i = \text{const} - \bar{a}_i$.

Покажемо, що така оптимальна стратегія краща не тільки будь-який чистої, але і будь-який змішаної. Нехай є змішана стратегія, яка представляє собою чергування чистих стратегій з імовірностями $S = (p_1, \dots, p_m)$, де $p_k = P(A_k)$ – імовірність використання k -ї стратегії. Тоді

$$\bar{a}_s = \sum_{k=1}^m p_k \bar{a}_k \leq \sum_{k=1}^m p_k \bar{a}_i = \bar{a}_i \sum_{k=1}^m p_k = \bar{a}_i, \quad (2.5.4)$$

де $\bar{a}_i$ (середнє) – це виграш при застосуванні оптимальної стратегії.

Вибір величин Q_j (імовірностей обстановки) при недостатності апіорної інформації можливий наступним чином:

- рівноймовірність варіантів обстановки $Q_j = 1/n$, де n – число варіантів обстановки
- ранжування (впорядкування за убутанням) ймовірностей варіантів обстановки і заміна невідомих ймовірностей на величини, що пропорційні членам спадної арифметичної прогресії

$$Q_1 : Q_2 : \dots : Q_n = n : (n-1) : \dots : 1, \quad (2.5.5)$$

$$Q_j = \frac{2(n-j+1)}{n(n+1)}, \quad \sum_{j=1}^n Q_j = \frac{2n \frac{n+1}{2}}{n(n+1)} = 1. \quad (2.5.6)$$

Критерії прийняття рішень, при невідомих ймовірностях обстановки.

Максимінний (мінімаксний) критерій Вальда.

При використанні цього критерію в якості оптимальної вибирається стратегія, при якій мінімальний виграш максимальний або максимальні витрати мінімальні, тобто стратегія, яка гарантує виграш не менше ніж $\alpha = \max_i \min_j a_{ij}$

чи витрати не більш ніж $\gamma = \min_i \max_j b_{ij}$. Це песимістичний критерій, що

орієнтується на гірші умови і вибирають в них стратегію, в якій виграш максимальний.

Таблиця 2.5.3. Приклад прийняття рішення за критерієм Вальда

$A_i \backslash B_j$	B_1	B_2	B_3	B_4
A_1	1	4	5	9
A_2	3	5	4	3
A_3	4	6	6	2

Наприклад, в табл. 2.5.3 для наданої матриці виграшів, жирним шрифтом виділені мінімальні виграші при застосуванні гравцем певної стратегії, а

жирним підкресленим шрифтом – максимальний з них, звідки випливає, що оптимальною стратегією за критерієм Вальда є A_2 .

Критерій азартного гравця

Як оптимальна обирається стратегія, яка дає найбільший виграш або найменші витрати в найкращих умовах: $\xi = \max_i \max_j a_{ij}$, $\zeta = \min_i \min_j b_{ij}$.

Таблиця 2.5.4. Приклад прийняття рішення за критерієм азартного гравця

$A_i \backslash B_j$	B_1	B_2	B_3	B_4
A_1	1	4	5	<u>9</u>
A_2	3	<u>5</u>	4	<u>3</u>
A_3	4	<u>6</u>	<u>6</u>	2

Оптимальною стратегією за критерієм азартного гравця є A_1 .

Критерій оптимізму – песимізму Гурвіца.

Якщо вхідною для прийняття рішення є матриця виграшів, то як оптимальна обирається стратегія, яка дає найбільше значення лінійної комбінації найбільших виграшів в найгірших та найкращих умовах

$$\mu = \max_i \left(k \cdot \min_j a_{ij} + (1-k) \cdot \max_j a_{ij} \right), \quad 0 \leq k \leq 1. \quad (2.5.7)$$

Якщо вхідною для прийняття рішення є матриця витрат, то як оптимальна обирається стратегія, яка дає найменше значення лінійної комбінації найменших витрат в найгірших та найкращих умовах

$$\eta = \min_i \left(k \cdot \max_j b_{ij} + (1-k) \cdot \min_j b_{ij} \right), \quad 0 \leq k \leq 1. \quad (2.5.8)$$

При $k = 1$ критерій Гурвіца переходить в критерій Вальда, а при $k = 0$ – в критерій азартного гравця.

Для матриці виграшів, яку представлено в табл. 2.5.3, оптимальною стратегією за критерієм Гурвіца при $k = 0,5$ є A_1 .

Критерій мінімального ризику Севіджа

В якості оптимальної вибирає стратегію, при якій максимальний ризик мінімальний. При цьому $r_{ij} = \beta_j - a_{ij} = \max a_{ij} - a_{ij}$ – недоотриманий виграш, $s_{ij} = b_{ij} - \min b_{ij}$ – перевищені витрати.

$$\varepsilon = \min_i \max_j r_{ij}, \quad \dots \quad \theta = \min_i \max_j S_{ij}. \quad (2.5.9)$$

Табл. 2.5.5 є таблицею ризиків, яка відповідає таблицям виграшів 2.5.3, 2.5.4.

Таблиця 2.5.5. Приклад прийняття рішення за критерієм Севіджа

$A_i \backslash B_j$	B_1	B_2	B_3	B_4
A_1	<u>3</u>	2	1	0
A_2	1	1	2	6
A_3	0	0	0	7

Оптимальною стратегією за критерієм Севіджа є A_1 .

Критерій Лапласа – Байєса

В якості оптимальної вибирає стратегію, що забезпечує максимальне середнє значення виграшу або мінімальне середнє значення витрат (при цьому приймається припущення про рівномірності умов).

$$\sigma = \max_i \frac{1}{n} \sum_{j=1}^n a_{ij}, \quad \psi = \min_i \frac{1}{n} \sum_{j=1}^n b_{ij}. \quad (2.5.10)$$

Іншими словами як оптимальну можна визначити стратегію, яка забезпечує максимальний сумарний виграш або мінімальні сумарні витрати за всіма варіантами обстановки.

Для матриці виграшів, яку представлено в табл. 2.5.3, оптимальною стратегією за критерієм Лапласа – Байєса є A_1 .

Редукція платіжних матриць. Метою редукції є зменшення розмірів матриць. При редукції з матриці виключаються стратегії гравців, які являються дублюючими або заздалегідь не вигідними для того чи іншого гравця. Надалі розглядаються платіжні матриці парних антагоністичних ігор. При цьому слід мати на увазі, що в таких іграх гравці переслідують протилежні цілі і кожна з сторін намагається отримати максимальний прибуток тобто нанести протилежній стороні максимальний збиток. Такі платіжні матриці прийнято записувати у вигляді виграшів гравця А або, відповідно програшів гравця В.

Нехай платіжна матриця має вигляд табл. 2.5.6.

Таблиця 2.5.6. Приклад вхідної платіжної матриці, яка піддається редукції

$A_i \backslash B_j$	B_1	B_2	B_3	B_4	B_5
A_1	-2	1	3	0	1
A_2	-1	-4	2	-1	-4
A_3	1	-5	6	3	-5

Проаналізуємо матрицю з позицій гравця В:

1. Значення програвів в стовпцях B_2 та B_5 співпадають, тому робимо висновок, що стратегії рівноцінні, дублюють одна одну і тому одну з них можемо виключити (виключимо B_5).

2. Стовпчик 3 приносить максимальний програш гравцю B при будь-яких стратегіях гравця A , адже $a_{i3} > a_{ij}$ при усіх i та $j \neq 3$, наприклад $a_{i3} > a_{i1}$.

Такий стовпчик-стратегію звать заздалегідь не вигідною, а стовпчики-стратегії B_1, B_2, B_4, B_5 суворо домінуючими. Стовпчик B_3 можна видалити, так як розумний гравець цю стратегію ніколи не використає.

3. При порівнянні стовпців B_1 та B_4 бачимо, що:

- $a_{14} > a_{11}$;
- $a_{24} = a_{21}$;
- $a_{34} > a_{31}$.

Тобто в стовпці B_4 містяться елементи, які строго більше елементів стовпця B_1 , або дорівнюють відповідним елементам B_1 .

В цьому випадку стовпчик/стратегія B_1 несуворо домінує над стовпчиком-стратегією B_4 . Останній можна виключити, оскільки ця стратегія у будь-якому разі приносить не менший програш гравцю B , ніж стратегія B_1 .

В результаті вилучення неефективних стратегій гравця B , гру розміру 3×5 , вдалося зменшити до розміру 3×2 (табл. 2.5.7).

Таблиця 2.5.7. Приклад вихідної платіжної матриці після редукції

$A_i \backslash B_j$	B_1	B_2
A_1	-2	1
A_2	-1	-4
A_3	1	-5

З аналогічних позицій у загальному випадку можна розглянути і ефективність стратегій гравця A , тобто виявити стратегії, що дублюються та суворо або не суворо домінуючі стратегії, але у випадку, що розглядається подальша редукція платіжної матриці неможлива.

Таким чином перш ніж починати рішення гри, слід за можливістю зменшити її розміри. Одним з прийомів зниження розмірів є використання принципу домінування.

Стратегія B_i домінує над стратегією B_j , якщо при будь-якій поведінці інших гравців використання стратегії B_i призводить до негіршого результату, ніж використання стратегії B_j .

Розрізняють суворе домінування, коли B_i дає більший виграш, ніж B_j в будь-яких умовах та несуворо, якщо при деяких діях інших гравців B_i забезпечує більший виграш, ніж B_j , а в іншому випадку – однаковий з нею.

Стратегія B_j зветься заздалегідь не вигідною, якщо для неї існує хоча б одна домінуюча стратегія.

Стратегія B_j є дублюючою стратегією B_i , якщо при будь-якій поведінці інших гравців використання стратегії B_j призводить до того ж результату, що і використання стратегії B_i .

При редукції платіжних матриць заздалегідь не вигідні та дублюючі стратегії підлягають вилученню.

Нижня та верхня ціни гри. Найважливішим питанням в теорії ігор є питання про вибір оптимальних стратегій для кожного з гравців.

Оптимальною стратегією гравця в матричній грі називається така, яка забезпечує йому максимальний виграш. Якщо гра повторюється неодноразово, то оптимальна стратегія повинна забезпечувати максимальний середній виграш.

При виборі цієї стратегії основою міркувань є припущення, що противник є так само розумний, як і власно гравець, і робить все, щоб добитися такої ж мети.

При цьому для вибору оптимальної стратегії використовують принцип максиміну: обирається та стратегія, щоб при найгіршій для гравця поведінці супротивника, гравець отримав максимальний виграш. Іншими словами, принцип максиміну передбачає вибір тієї стратегії, при якій мінімальний виграш гравця для будь-яких стратегій супротивника є максимальним.

Як видно, принцип максиміну – це принцип, який розрахований на песимістичний сценарій гри.

Визначимо найкращу стратегію гравця A з урахуванням всіх можливих відповідей на неї гравця B . При цьому слід розраховувати на те, що на будь-яку стратегію A_i , гравця A гравець B відповість стратегією B_j , для якої виграш гравця A виявиться мінімальним, оскільки гравець B намагається зашкодити гравцю A .

Алгоритм знаходження максиміну (мінімаксу):

1. В рядку платіжної матриці, що відповідає стратегії A_i , знайти мінімальне з чисел

$$a_i = \min_j a_{ij}, \quad j = 1, \dots, n. \quad (2.5.11)$$

Це гарантований виграш гравця A при застосуванні стратегії A_i . Очевидно, що гравцю A вигідно вибирати таку стратегію A_i для якої значення гарантованого виграшу було б найбільшим.

2. Серед всіх значень a_i обрати найбільше число, яке визначається за формулою

$$\alpha = \max_i a_i = \max_i \min_j a_{ij}, \quad j = 1, \dots, n, \quad i = 1, \dots, m \quad (2.5.12)$$

і зветься нижньою ціною гри або максиміном.

Максимін – це максимальний виграш, який гравець A може собі гарантувати в грі проти розумного противника.

Якщо гравець A буде дотримуватись максимінної стратегії, то йому при будь-якій розумній поведінці гравця B гарантовано виграш, не менший ніж α . Стратегія, яка відповідає максиміну називається максимінною.

3. В стовпці платіжної матриці, що відповідає стратегії B_j знайти максимальне з чисел

$$b_j = \max_i a_{ij}, \quad i = 1, \dots, m. \quad (2.5.13)$$

Це гарантований програш гравця B при застосуванні стратегії B_j – найгірший з програшів. Очевидно, що гравець B намагається перетворити виграш гравця A в мінімальний, тобто він повинен обрати стратегію, яка дає найменший програш.

Серед всіх значень b_j обрати найбільше число, яке визначається за формулою

$$\beta = \min_j b_j = \min_j \max_i a_{ij}, \quad i = 1, \dots, m, \quad j = 1, \dots, n \quad (2.5.14)$$

і зветься верхньою ціною гри або мінімаксом.

Мінімакс – це мінімальний програш, який гравець B може собі гарантувати в грі проти розумного противника.

Якщо гравець B буде дотримуватись найбільш обережної з усіх стратегій – мінімаксною, то йому при будь-якої випадку забезпечено програш, не більший ніж β .

Стратегія, яка відповідає мінімаксу називається мінімаксною.

Приклад. Нижче наведено платіжну матрицю парної гри “в три пальці”. В ній гравці A та B одночасно та незалежно один від одного показують 1 або 2 або 3 пальці. Виграш визначається загальною кількістю показаних пальців. Якщо це число парне то B платить це значення на користь A , якщо непарне, то A платить на користь B . Тобто елементи платіжної матриці – це виграші A , а якщо вони взяти зі знаком мінус – виграші B .

Таблиця 2.5.8. Платіжні матриці гри “в три пальці” з ілюстрацією визначення максимінної та мінімаксною стратегій

$A_i \backslash B_j$	B_1	B_2	B_3
A_1	2	<u>-3</u>	4
A_2	-3	4	<u>-5</u>
A_3	4	<u>-5</u>	6

$A_i \backslash B_j$	B_1	B_2	B_3
A_1	2	-3	4
A_2	-3	4	-5
A_3	4	-5	<u>6</u>

При цьому нижня та верхня ціни гри складають

$$\alpha = \max_i a_i = \max_i \min_j a_{ij} = \max_i \{-3; -5; -5\} = -3, \quad j = 1, \dots, 3, \quad i = 1, \dots, 3$$

$$\beta = \min_j b_j = \min_j \max_i a_{ij} = \min_j \{4; 4; 6\} = 4, \quad i = 1, \dots, 3, \quad j = 1, \dots, 3$$

Максимінною стратегією гравця A є A_1 . Мінімаксною стратегією гравця B є B_1 та B_2 . В табл. 2.5.8 вони виділені жирним шрифтом.

Сідлова точка. В теорії ігор принцип обережності, який рекомендує гравцям дотримання максимінної і мінімаксної стратегій, називається принцип мінімаксу. Він впливає з припущення про обережність гравців, тобто з бажання розв'язати конфліктну ситуацію найкращим чином для всіх учасників конфлікту.

Фактичний виграш гравця A (програш гравця B) при розумних діях партнерів обмежений верхньої та нижньої ціною гри. Якщо ж ці вирази рівні, тобто

$$v = \max_i \min_j a_{ij} = \min_j \max_i a_{ij}, \quad (2.5.15)$$

то такий варіант називається грою з сідловою точкою. В матриці гри з сідловою точкою існує принаймні один елемент, який є одночасно мінімальним у своєму рядку і максимальним у своєму стовпчику; такий елемент і називають сідловою точкою.

Сідловій точці відповідає пара максимінної і мінімаксної стратегій, ці стратегії зводяться оптимальними, а їх сукупність – розв'язком гри.

Приклад. Нижче наведено платіжну матрицю парної гри, де гравець A – протиповітряна оборона (ППО), її чиста стратегія – вид засобу ППО, гравець B – військово-повітряні сили, їх чиста стратегія – тип літального апарату. Елемент платіжної матриці – ймовірність ураження літального апарату засобом ППО p . Виграш гравця A при цьому складає p , а гравця B – $1-p$.

Таблиця 2.5.9. Платіжні матриці гри з сідловою точкою з ілюстрацією визначення максимінної та мінімаксної стратегій

$A_i \backslash B_j$	B_1	B_2	B_3
A_1	0,5	0,6	0,8
A_2	0,9	0,7	0,8
A_3	0,7	0,5	0,6

$A_i \backslash B_j$	B_1	B_2	B_3
A_1	0,5	0,6	0,8
A_2	0,9	0,7	0,8
A_3	0,7	0,5	0,6

При цьому нижня та верхня ціни збігаються і складають

$$\alpha = \max_i a_i = \max_i \min_j a_{ij} = \max_i \{0,5; 0,7; 0,5\} = 0,7, \quad j = 1, \dots, 3, \quad i = 1, \dots, 3$$

$$\beta = \min_j b_j = \min_j \max_i a_{ij} = \min_j \{0,9; 0,7; 0,8\} = 0,7, \quad i = 1, \dots, 3, \quad j = 1, \dots, 3$$

Так a_{22} є сідловою точкою, гра має рішення у чистих стратегіях A_2 та B_2 . Значення платіжної матриці одночасно є мінімальним по рядку і максимальним по стовпцю.

Рішення ігор у змішаних стратегіях. Якщо гра не має сідлової точки, то застосування чистих стратегій не дасть оптимального рішення гри. В такому випадку можна отримати оптимальне рішення випадковим чином, чергуючи чисті стратегії.

Змішаною стратегією S_A гравця A зветься застосування чистих стратегій $A_1, A_2, \dots, A_m$ з ймовірностями $p_1, p_2, \dots, p_m$, при чому сума ймовірностей дорівнює 1

$$\sum_{i=1}^m p_m = 1. \quad (2.5.15)$$

Змішану стратегію гравця A записують у вигляді матриці або у вигляді рядка

$$S_A = \begin{pmatrix} A_1 & A_2 & \dots & A_m \\ p_1 & p_2 & \dots & p_m \end{pmatrix}, \quad S_A = (p_1 \quad p_2 \quad \dots \quad p_m). \quad (2.5.16)$$

Аналогічно записують змішану стратегію S_B гравця B

$$S_B = \begin{pmatrix} B_1 & B_2 & \dots & B_n \\ q_1 & q_2 & \dots & q_n \end{pmatrix}, \quad S_B = (q_1 \quad q_2 \quad \dots \quad q_n), \quad (2.5.17)$$

де сума ймовірностей застосування стратегій дорівнює 1

$$\sum_{j=1}^n q_n = 1. \quad (2.5.18)$$

На підставі принципу мінімаксу визначається оптимальне рішення гри: пара оптимальних стратегій S_A, S_B в загальному випадку змішаних, що мають наступну властивість: якщо один з гравців дотримується своєї оптимальної стратегії, то другому не може бути вигідно відступати від своєї.

Виграш, відповідний оптимальному рішення називається ціною гри ν , іноді його також звать чистою ціною гри.

Ціна гри задовольняє нерівності $\alpha \leq \nu \leq \beta$, де α та β – нижня та верхня ціни гри.

Основна теорема теорії ігор або теорема Неймана формулюється наступним чином: кожна кінцева гра має принаймні одне оптимальне рішення, можливо серед змішаних стратегій.

Теорема про активні стратегії: якщо один з гравців дотримується своєї оптимальної стратегії, то виграш залишається незмінним та дорівнює ціні гри ν , якщо другий гравець не виходить за межі своїх активних стратегій.

Ця теорема дозволяє визначати оптимальні стратегії при відсутності сідлової точки.

Розглянемо гру розміром 2×2 (найпростіший випадок кінцевої гри).

Якщо сідлова точка існує, то оптимальне рішення – це пара чистих стратегій, відповідних цій сідловій точці.

Якщо сідлової точки не існує, то за теоремою Неймана оптимальне рішення визначається парою змішаних стратегій $S_A=(p_1,p_2)$ та $S_B=(q_1,q_2)$.

Згідно теореми про активні стратегії якщо гравець A дотримується своєї оптимальної стратегії S_A , то його середній виграш дорівнює ціні гри v , яку б активну стратегію не використовував гравець B .

Виграш гравця A (програш гравця B) – випадкова величина, математичне очікування якої є ціною гри.

Нехай гра задана платіжною матрицею

$$P = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}.$$

Середній виграш гравця A , якщо він використовує оптимальну змішану стратегію $S_A=(p_1,p_2)$, гравець B – чисту стратегію B_1 (тобто перший стовпчик матриці P), дорівнює ціні гри: $a_{11}p_1 + a_{21}p_2 = v$.

Той же середній виграш отримує гравець A , якщо гравець B використовує стратегію B_2 (тобто другий стовпчик матриці P): $a_{12}p_1 + a_{22}p_2 = v$.

Враховуючи, що сума ймовірностей дорівнює 1, можна отримати систему рівнянь для визначення стратегії $S_A=(p_1,p_2)$ та ціни гри v

$$\begin{cases} a_{11}p_1 + a_{21}p_2 = v \\ a_{12}p_1 + a_{22}p_2 = v \\ p_1 + p_2 = 1 \end{cases} \quad (2.5.19)$$

Вирішуючи систему отримуємо ймовірності, що визначають оптимальну стратегію, та ціну гри

$$p_1 = \frac{a_{22} - a_{21}}{a_{11} + a_{22} - a_{12} - a_{21}}; \quad p_2 = \frac{a_{11} - a_{12}}{a_{11} + a_{22} - a_{12} - a_{21}}; \quad v = \frac{a_{11}a_{22} - a_{12}a_{21}}{a_{11} + a_{22} - a_{12} - a_{21}}. \quad (2.5.20)$$

При знаходженні $S_B=(q_1,q_2)$ – оптимальної стратегії гравця B використовується те, що при будь-якій чистій стратегії гравця A (A_1 або A_2) середній програш гравця B дорівнює ціні гри v

$$\begin{cases} a_{11}q_1 + a_{12}q_2 = v \\ a_{21}q_1 + a_{22}q_2 = v \\ q_1 + q_2 = 1 \end{cases} \quad (2.5.21)$$

Вирішуючи систему отримуємо ймовірності, що визначають оптимальну стратегію

$$q_1 = \frac{a_{22} - a_{12}}{a_{11} + a_{22} - a_{12} - a_{21}}; \quad q_2 = \frac{a_{11} - a_{21}}{a_{11} + a_{22} - a_{12} - a_{21}}. \quad (2.5.22)$$

Приклад. Знайти рішення гри 2х2 з платіжною матрицею

$$P = \begin{pmatrix} 8 & 5 \\ 4 & 6 \end{pmatrix}.$$

Перевіримо наявність сідлової точки.

$$\alpha = \max_i a_i = \max_i \min_j a_{ij} = \max_i \{5; 4\} = 5, \quad j = 1, 2, \quad i = 1, 2.$$

$$\beta = \min_j b_j = \min_j \max_i a_{ij} = \min_j \{8; 6\} = 6, \quad i = 1, 2, \quad j = 1, 2.$$

Нижня і верхня ціни гри не збігаються, сідлова точка відсутня і гра має рішення у змішаних стратегіях. Максимінною стратегією гравця $A \in A_1$. Мінімаксною стратегією гравця $B \in B_2$.

Система рівнянь для визначення ймовірностей застосування стратегій гравцем A має вигляд

$$\begin{cases} 8p_1 + 4p_2 = \nu \\ 5p_1 + 6p_2 = \nu \\ p_1 + p_2 = 1 \end{cases}$$

Її рішення $p_1=0,4$, $p_2=0,6$, $\nu=5,6$.

Система рівнянь для визначення ймовірностей застосування стратегій гравцем B має вигляд

$$\begin{cases} 8q_1 + 5q_2 = \nu \\ 4q_1 + 6q_2 = \nu \\ q_1 + q_2 = 1 \end{cases}$$

Її рішення $q_1=0,2$, $q_2=0,8$, $\nu=5,6$.

Постановка задачі

Дано

Платіжні матриці гри зі стихією та парної антагоністичної гри.

Потрібно

Визначити оптимальні стратегії гравців. Для парної антагоністичної гри також визначити максимінну та мінімаксу стратегію, нижню, верхню та чисту ціни гри.

Рекомендації щодо виконання роботи

1. Вважаючи, що платіжна матриця гри зі стихією для варіантів 1 – 14 є матрицею витрат, а для варіантів 15 – 28 – матрицею виграшів, визначити оптимальні стратегії гравця за відомим імовірнісним розподілом випадкової обстановки, та за критеріями Вальда, азартного гравця, Гурвіца при $k = 0,5$, Севіджа та Лапласа – Байєса.
2. Виконати редукцію платіжної матриці парної антагоністичної гри шляхом вилучення заздалегідь не вигідної стратегії. Визначити максимінну та мінімаксу стратегію гравців, нижню, верхню ціни гри. Переконалися, що гра не має сідлової точки.
3. Скласти і розв'язати систему рівнянь для визначення ймовірнісних розподілів оптимальних стратегій гравців та чистої ціни гри.

Вимоги до звіту

Звіт може оформлюватися як “вручну” так і з застосуванням програмних засобів. В першому випадку у якості звіту надаються фотокопії розрахунків, у другому – екранні копії або їх програмні коди.

Звіт по роботі повинен містити:

1. Назву дисципліни та лабораторної роботи.
2. Прізвище, ім'я та по батькові здобувача вищої освіти, код групи, номер варіанту завдання.
3. Окремі платіжні матриці гри зі стихією для кожного з критеріїв згідно п.1 рекомендацій щодо виконання роботи. Їх оформлення повинно відповідати оформленню табл. 2.5.3 – 2.5.5.
4. Вхідну та редуційовану платіжні матриці парної антагоністичної гри з обґрунтуванням виключення певної стратегії одного з гравців.
5. Редуційовані платіжні матриці парної антагоністичної гри з ілюстрацією визначення максимінної та мінімаксної стратегій. Їх оформлення повинно відповідати оформленню табл. 2.5.8 – 2.5.9.
6. Систему рівнянь для визначення ймовірнісних розподілів оптимальних стратегій гравців та чистої ціни гри.

Контрольні питання і завдання

1. Що є предметом теорії статистичних рішень і теорії антагоністичних ігор?
2. Що таке гра?
3. Які види ходів розрізняють у іграх? Приведіть приклади ігор, які містять ходи різних видів.
4. Що регламентують правила, яким підпорядковується гра?
5. Поясніть різницю між чистими та змішаними стратегіями.
6. Яка стратегія зветься оптимальною?

7. Поясніть яким чином можуть бути отримані кількісні оцінки ймовірностей варіантів випадкової обстановки за наявності їх ранжованої послідовності.
8. Сформулюйте порядок отримання матриці ризиків з платіжної матриці.
9. Запишіть формули для величин, які максимізуються або мінімізуються при прийнятті рішень за допомогою критеріїв: Вальда, азартного гравця, Гурвіца, Севіджа, Лапласа Байеса.
10. Вважаючи, що в табл. 2.5.3 представлена матриця витрат, знайдіть оптимальні стратегії за критеріями: Вальда, азартного гравця, Гурвіца, Севіджа, Лапласа Байеса.
11. Дайте визначення стратегій: суворо та несуворо домінуючих, заздалегідь не вигідних та дублюючих. Які з них підлягають вилученню з платіжної матриці при редукції?
12. Дайте визначення максимінної та мінімаксної стратегій.
13. Дайте визначення нижньої та верхньої цін гри.
14. Дайте визначення сідлової точки платіжної матриці.
15. Сформулюйте основну теорему теорії ігор та теорему про активні стратегії.
16. Дайте визначення ціни гри.
17. Викладіть порядок рішення парної кінцевої гри в змішаних стратегіях.

Варіанти завдань

Містяться у додатку 5.

2.6 Лабораторна робота № 6

Прийняття рішень методами кластерного аналізу

Об'єкт – методи кластерного аналізу. Предмет – ієрархічні агломеративні методи і методи ітеративного угруповання кластерного аналізу. Мета – кластеризація вибірки результатів діагностики хвороби за значеннями факторних ознак (симптомів) для прийняття рішення про необхідність оперативного втручання.

Стислі теоретичні відомості

Класичними методами класифікації без навчання є методи кластерного аналізу (таксономії). За їх допомогою вирішують проблему такого розбиття (кластеризації) множини об'єктів, за якого всі об'єкти, що належать до одного класу, були б більш подібними один до одного, ніж до об'єктів інших класів.

Розглянемо математичну постановку завдання кластерного аналізу. Припустимо, що існує множина об'єктів, що підлягають класифікації $I = \{\bar{I}_j\}_{j=1, \dots, n}$.

Кожний об'єкт характеризується множиною ознак $\{C_i\}_{i=1, \dots, p}$. Ознаки можуть бути як не випадковими так і випадковими величинами та вимірюваними в будь-яких шкалах.

Як правило, перед процедурою кластерного аналізу застосовується стандартизація даних – значень i -ї ознаки для j -го об'єкту ξ_{ji} згідно співвідношення

$$x_{ji} = \frac{\xi_{ji} - \bar{\xi}_i}{\sqrt{D_i}}. \quad (2.6.1)$$

Тут $\bar{\xi}_i$, D_i – відповідно середнє значення та статистична дисперсія i -ї ознаки.

Це дозволяє перейти до нормованих та безрозмірних величин. Стандартизовані результати виміру i -ї ознаки для j -го об'єкту задаються матрицею спостережень $\{x_{ji}\}_{j=1,\dots,n, i=1,\dots,p}$, а сукупність всіх ознак для кожного з об'єктів – множиною векторів

$\mathbf{X} = \{\bar{X}_j\}_{j=1,\dots,n}$. Нехай m – ціле число, $m < n$, яке може бути заздалегідь невідоме.

Завдання кластерного аналізу полягає в тому щоб на основі даних про величини ознак об'єктів розбити множину об'єктів на m кластерів $\{\pi_k\}_{k=1,\dots,m}$, при виконанні умов

$$\begin{cases} \pi_k \neq \emptyset \quad \forall k \\ \pi_{k1} \cap \pi_{k2} = \emptyset \quad \forall k1 \neq k2 \\ \bigcup_{k=1}^m \pi_k = \mathbf{I} \end{cases} \quad (2.6.2)$$

При цьому всі об'єкти, що належать до одного кластеру, повинні бути як можна більш схожі між собою, а ті що належать до різних кластерів – більш різні між собою.

Розв'язанням завдання кластерного аналізу є розбиття множини об'єктів на кластери, яке оптимізує цільову функцію, що враховує внутрішньокластерні та міжкластерні міри збіжності. Прикладом такої функції може виступати сума по всіх кластерах внутрішньокластерних сум квадратів відхилень ознак об'єктів від середніх за кластером. Якщо j -й об'єкт характеризується скалярною ознакою x_j , ця функція буде мати вигляд

$$W = \sum_{k=1}^m \sum_{I_j \in \pi_k} \left(x_j - \frac{1}{n_k} \sum_{I_i \in \pi_k} x_i \right)^2. \quad (2.6.3)$$

Тут n_k – число об'єктів в кластері π_k , $n = \sum_{k=1}^m n_k$.

Міри відстаней і подібностей. Для формування кластерів застосовують міри відмінності серед яких найбільше поширення отримали міри типу “відстань”. При їх застосуванні об'єкти вважають тим більш подібними один до одного, чим меншою є відстань між ними. Як міру відстані (метрику) можна використовувати будь-яку функцію $\rho(\vec{X}_r, \vec{X}_v)$, що визначена на множині $\mathbf{X} = \{\vec{X}_j\}_{j=1, \dots, n}$ і задовольняє таким вимогам:

- $\rho(\vec{X}_r, \vec{X}_v) \geq 0 \quad \forall r, v \in \mathbf{X}$;
- $\rho(\vec{X}_r, \vec{X}_v) = 0 \Leftrightarrow \vec{X}_r = \vec{X}_v$;
- $\rho(\vec{X}_r, \vec{X}_v) = \rho(\vec{X}_v, \vec{X}_r) \quad \forall r, v \in \mathbf{X}$ (рефлексивність);
- $\rho(\vec{X}_r, \vec{X}_v) \leq \rho(\vec{X}_r, \vec{X}_w) + \rho(\vec{X}_w, \vec{X}_v) \quad \forall r, v, w \in \mathbf{X}$ (правило трикутника).

Відстані між всіякими парами множини з n об'єктів зручно описувати матрицею відстаней $P = \{\rho_{rv}\}_{r=1, \dots, n, v=1, \dots, n}$ розміром $n \times n$, яка має наступні властивості:

1. Елементи головної діагоналі є нульовими $\rho_{rr} = 0$;
2. Матриця є симетричною $\rho_{rv} = \rho_{vr}$ або $P = P^T$.

Вибір міри відстані істотно впливає на результат класифікації. У якості міри для кількісних ознак найчастіше використовують евклідову відстань

$$\rho(\vec{X}_r, \vec{X}_v) = \sqrt{\sum_{i=1}^p (x_{ri} - x_{vi})^2} \quad (2.6.4)$$

Приклад. Нехай вектора ознак двох об'єктів $\vec{X}_r = (1, 0, -1, 2)$; $\vec{X}_v = (3, 3, -2, -1)$. Евклідова відстань між ними становитиме

$$\rho(\vec{X}_r, \vec{X}_v) = \sqrt{(1-3)^2 + (0-3)^2 + (-1+2)^2 + (2+1)^2} = \sqrt{4+9+1+9} = \sqrt{23}.$$

Поряд з евклідовою використовуються такі міри відстані: зважена евклідова

$$\rho(\vec{X}_r, \vec{X}_v) = \sqrt{\sum_{i=1}^p w_i \cdot (x_{ri} - x_{vi})^2} , \quad (2.6.5)$$

де $0 \leq w_i \leq 1$ – вага i -ї ознаки, $\sum_{i=1}^p w_i = 1$;

манхеттенська відстань (city block, l_1 -норма),

$$\rho(\vec{X}_r, \vec{X}_v) = \sum_{i=1}^p |x_{ri} - x_{vi}| , \quad (2.6.6)$$

l_p -норма,

$$\rho(\vec{X}_r, \vec{X}_v) = \sqrt[p]{\sum_{i=1}^p |x_{ri} - x_{vi}|^p} , \quad (2.6.7)$$

супремум-норма,

$$\rho(\vec{X}_r, \vec{X}_v) = \sup_i \{ |x_{ri} - x_{vi}| \} , \quad (2.6.8)$$

Відстань Махаланобіса

$$\rho(\vec{X}_r, \vec{X}_v) = \sqrt{(\vec{X}_r - \vec{X}_v)^T \cdot S^{-1} \cdot (\vec{X}_r - \vec{X}_v)} , \quad (2.6.9)$$

де S – коваріаційна матриця.

Іноді замість міри відстані застосовується міра подібності яка задовольняє вимогам:

- $0 \leq S(\vec{X}_r, \vec{X}_v) \leq 1 \quad \forall r, v \in \mathbf{X}$;
- $S(\vec{X}_r, \vec{X}_v) = 1 \Leftrightarrow \vec{X}_r = \vec{X}_v$;
- $S(\vec{X}_r, \vec{X}_v) = S(\vec{X}_v, \vec{X}_r) \quad \forall r, v \in \mathbf{X}$ (рефлексивність).

Міри подібності між всілякими парами множини з n об'єктів зручно описувати матрицею подібностей $\mathbf{S} = \{S_{rv}\}_{r=1, \dots, n, v=1, \dots, n}$ розміром $n \times n$, яка має наступні властивості:

1. Елементи головної діагоналі є одиничними $S_{rr} = 0$;

2. Матриця є симетричною $S_{rv} = S_{vr}$ або $S = S^T$.

Для порядкових ознак використовуються коефіцієнти рангової кореляції Спірмена й Кендалла. Для номінальних та дихотомічних ознак застосовуються коефіцієнт спряженості Чупрова, коефіцієнти асоціації та колігації Юла, коефіцієнт спряженості Бравайса. Розглянуті показники можна перетворити у відстані, віднімаючи обчислені значення від одиниці.

Стислий огляд методів кластерного аналізу. Залежно від кількості вихідних спостережень виділяють задачі кластеризації невеликих за обсягом (до декількох десятків об'єктів) масивів спостережень і задачі кластеризації великих масивів. Такий поділ зумовлений різницею методів, які доцільно використовувати при кластеризації відповідних даних.

З погляду апріорної інформації про кількість кластерів вирізняють такі типи задач:

- із заданою кількістю класів;
- з невідомою кількістю класів, яку треба оцінити;
- з невідомою кількістю класів, яку не потрібно оцінювати (таку задачу зазвичай формулюють як побудову ієрархічного дерева, або дендрограми вхідної сукупності).

Найпоширенішими методами кластерного аналізу є:

- метод повного перебирання;
- ієрархічні агломеративні та дивізімні методи (ближнього зв'язку, середнього зв'язку Кінга, Уорда, далекого зв'язку);
- ієрархічні дивізімні методи;
- ітеративні методи групування (метод К-середніх (K-means) Мак-Куїна);
- алгоритми типу розрізування графа (кореляційних плеяд Терентьева, вроцлавська таксономія);
- метод пошуку локальних згущень;
- факторні методи.

Окремо слід виділити методи нечіткої кластеризації в результаті застосування яких об'єкти можуть з певними ймовірностями потрапляти до декількох кластерів (метод нечітких С-середніх (Fuzzy C-means)).

Метод повного перебирання. Полягає у повному перебиранні всіх можливих варіантів розбиття множини об'єктів на кластери, визначення значення цільової функції для кожного з варіантів і вибір оптимального серед них. Метод застосовується при невеликій кількості об'єктів, що кластеризуються. Число варіантів розбиття n об'єктів на m непорожніх підмножин (число варіантів кластеризації) визначається числом Стірлінга II роду і дорівнює

$$N = \frac{1}{m!} \sum_{j=0}^m C_m^j (-1)^j (m-j)^n \quad (2.6.10)$$

Приклад. Число варіантів кластеризації $n=4$ об'єктів на $m=2$ кластери становить

$$N = \frac{1}{2!} \left(C_2^0 \cdot (-1)^0 \cdot (2-0)^4 + C_2^1 \cdot (-1)^1 \cdot (2-1)^4 + C_2^2 \cdot (-1)^2 \cdot (2-2)^4 \right) = \frac{1}{2} (16 - 2) = 7$$

Можливий склад кластерів: (123),(4); (124),(3); (134),(2); (234),(1); (12),(34); (13),(24); (14),(23).

Ієрархічні агломеративні методи. Призначені переважно для побудови ієрархічних дерев відносно невеликих за обсягом сукупностей. Іноді їх використовують також для задач із заданою кількістю класів, або з невідомою кількістю класів, яку треба оцінити. У цьому випадку реалізацію ієрархічного алгоритму продовжують до досягнення кількості класів, яка дорівнює заздалегідь заданому числу, або до досягнення екстремуму одного з критеріїв якості розбиття.

Перевагами ієрархічних методів є можливість більш повного і тонкого аналізу структури досліджуваної сукупності порівняно з іншими методами, а також наочність подання результатів кластеризації. Їх основними недоліками є громіздкість обчислювальної процедури, яка пов'язана з перерахунком усієї матриці відстаней на кожному кроці.

Метод ближнього зв'язку є найпростішим для розуміння з ієрархічних агломеративних методів кластерного аналізу. Процес в цьому випадку починають з пошуку та об'єднання двох найближчих один до одного об'єктів у матриці відстаней.

На наступному етапі знаходять два наступні найближчі об'єкти й так само до повного вичерпання матриці відстаней. Як правило, робота алгоритму закінчується, коли всі спостереження об'єднані в один кластер. Для відокремлення кластерів після закінчення кластеризації задають пороговий рівень відстані, на якому можна виділити більше, ніж один кластер.

У методі ближнього зв'язку два об'єкти потрапляють до одного й того самого кластера в тому випадку, коли існує ланцюжок близьких один до одного об'єктів, які їх з'єднують. Іноді це призводить до необґрунтованого зарахування об'єктів до одного й того самого кластера (ланцюжковий ефект). У процесі кластеризації можна явно простежити утворення таких ланцюжків. Для запобігання цьому ефекту можна задавати обмеження на максимальну відстань між елементами одного кластера. Кластери, одержувані за методом ближнього зв'язку, не обов'язково бувають опуклими. Залежно від обставин, це можна розглядати і як перевагу, і як недолік методу.

Результати ієрархічних методів кластерного аналізу стають більш наочними, якщо їх подати у вигляді вертикальних або горизонтальних дендрограм. Приклад вертикальної дендрограми наведено на рис. 1. Тут по горизонталі

відкладаються номери об'єктів, що кластеризуються, а по вертикалі – значення відстаней. Вертикальні лінії дендрограми відповідають об'єктам та кластерам, а горизонтальними відображається процес об'єднання об'єктів та кластерів нижнього рівня у кластери верхнього рівня. Об'єкти на горизонтальній осі прийнято розташовувати таким чином, щоб дендрограми не мали самоперетинів. За допомогою дендрограми легко визначити склад кластерів для будь-якого їх числа m такого, що $1 < m < n$. Для цього треба знайти рівень відстані, якому відповідає горизонтальна пряма, така що кількість її перетинів з вертикальними лініями дендрограми – кластерами дорівнює m . До складу певного отриманого кластеру входять всі об'єкти, що розташовуються у гілках дендрограми нижче точки перетину, яка визначає кластер.

Приклад. Для заданої матриці відстаней між $n=6$ об'єктами потрібно провести кластеризацію ієрархічним агломеративним методом ближнього зв'язку, побудувати дендрограму, визначити кількість та склад кластерів для різних значень відстані об'єднання.

Таблиця 2.6.1. Вхідна матриця відстаней

Номер об'єкте	1	2	3	4	5	6
	0	0,5	0,1	0,5	0,5	0,3
		0	0,5	0,4	0,6	0,5
			0	0,5	0,5	0,4
				0	0,2	0,5
					0	0,5
						0

Найближчими один до одного у матриці відстаней є об'єкти 1 та 3. Вони об'єднуються у кластер на відстані 0,1. Два наступні найближчі об'єкти 4 та 5 об'єднуються у окремий кластер на відстані 0,2. На відстані 0,3 між об'єктами 1 та 6 утворюється кластер (1,3,6), на відстані 0,4 між об'єктами 2 та 4 утворюється кластер (2,4,5) і на відстані 0,5 відбувається повне об'єднання. Дендрограма, яка ілюструє описаний процес, представлена на рис. 2.6.1.

Метод середнього зв'язку Кінга подібний до методу ближнього зв'язку. Його відмінність полягає в тому, що об'єднані до одного кластера об'єкти надалі вважають одним об'єктом з усередненими за кластером параметрами. В іншому варіанті методу середнього зв'язку відстань між кластерами розраховують як середнє значення відстаней між усіма можливими парами представників цих кластерів. При використанні методу середнього зв'язку в процесі кластеризації також простежується формування ланцюжків об'єктів.

Метод Уорда відрізняється від методу середнього зв'язку тим, що підставою для приєднання об'єкта до кластера є не близькість у значенні певної міри відстані, а мінімум дисперсії всередині кластера після поміщення до нього обраного об'єкта.

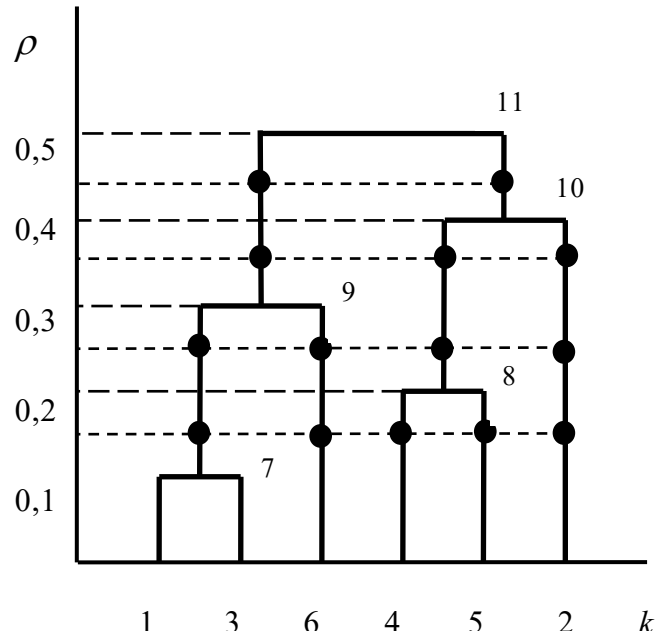


Рис. 2.6.1. Вертикальна дендрограма

Формула Ланса і Уільямса дозволяє описати правила групування для будь-якого ієрархічного агломеративного метода

$$\rho(h, k) = A(i) \cdot \rho(h, i) + A(j) \cdot \rho(h, j) + B \cdot \rho(i, j) + C \cdot |\rho(h, i) - \rho(h, j)|. \quad (2.6.11)$$

Тут $\rho(h, k)$ – відстань між кластерами h та k , при чому кластер k є результатом об'єднання кластерів (або об'єктів i та j). За допомогою цієї формули можна обрахувати значення відстані $\rho(h, k)$ на якій відстані відбуватиметься чергове об'єднання.

Зокрема для методу ближнього зв'язку $A(i) = A(j) = 0,5$, $B = 0$, $C = -0,5$.

Для розглянутого вище прикладу утворення кластеру (1,3,6) відбувається на відстані

$$\begin{aligned} \rho(6, (1, 3)) &= A(1) \cdot \rho(6, 1) + A(3) \cdot \rho(6, 3) + C \cdot |\rho(6, 1) - \rho(6, 3)| = \\ &= 0,5 \cdot 0,3 + 0,5 \cdot 0,4 - 0,5 \cdot |0,3 - 0,4| = 0,3. \end{aligned}$$

а кластеру (2,4,5) – на відстані

$$\begin{aligned} \rho(2, (4, 5)) &= A(4) \cdot \rho(2, 4) + A(5) \cdot \rho(2, 5) + C \cdot |\rho(2, 4) - \rho(2, 5)| = \\ &= 0,5 \cdot 0,4 + 0,5 \cdot 0,6 - 0,5 \cdot |0,4 - 0,6| = 0,4. \end{aligned}$$

Метод K-means Мак-Куїна. Цей метод відноситься до класу ітеративних методів групування. Вхідними даними для алгоритму є матриця спостережень $X = \{x_{ji} \mid \substack{j=1, \dots, n \\ i=1, \dots, p}\}$ та кількість кластерів, що формується. m . Графічна схема алгоритму кластеризації за методом K-means Мак-Куїна представлена на рис. 2.6.2.

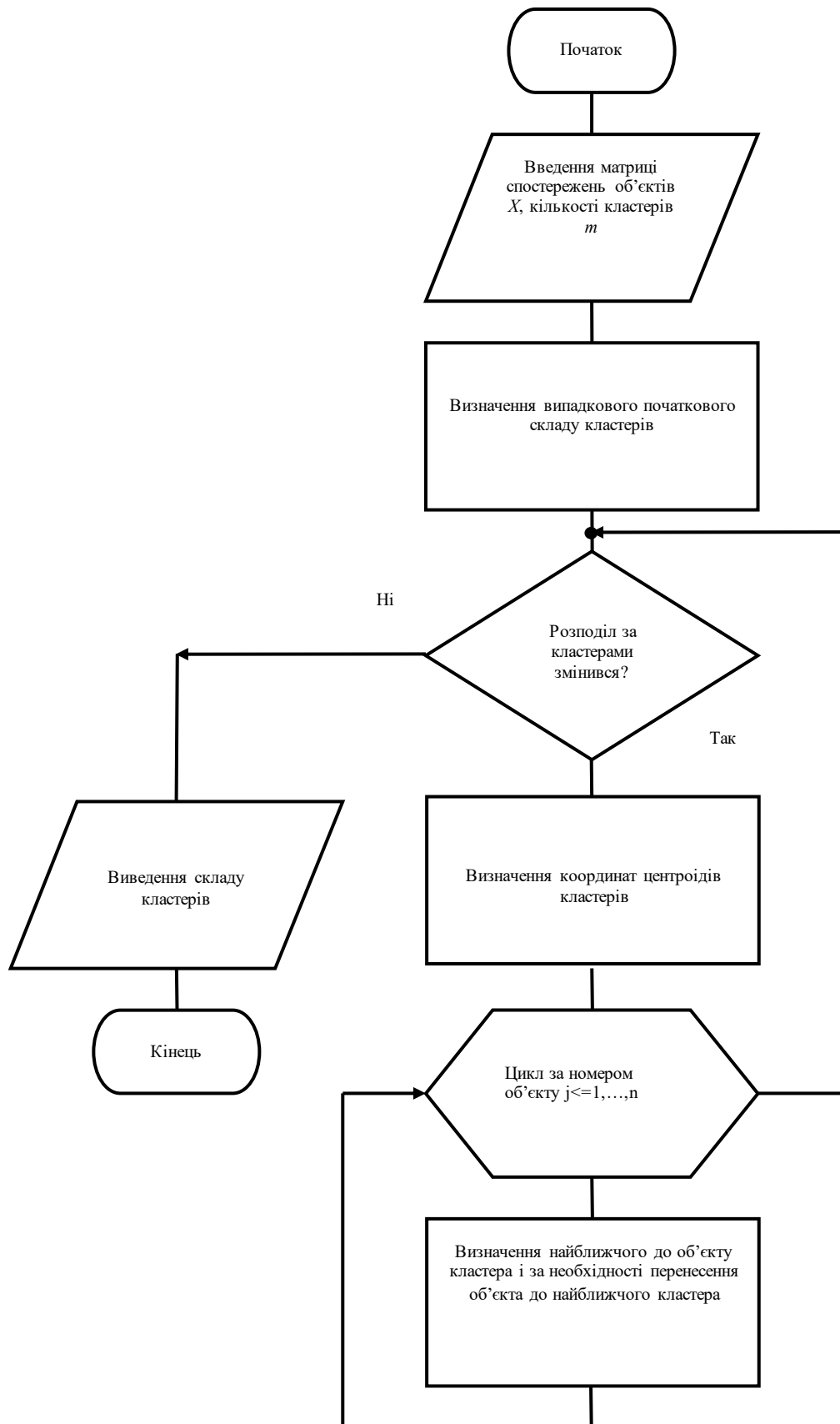


Рис. 2.6.2. Графічна схема алгоритму кластеризації за методом K- means Мак-Куїна

Початкове розбиття об'єктів за кластерами є довільним і може бути випадковим. Координати центроїдів кластеру π_k визначаються як

$$\bar{x}_{ki} = \frac{1}{n_k} \sum_{I_j \in \pi_k} x_{ji} . \quad (2.6.12)$$

Тут n_k – число об'єктів в кластері π_k . За звичай графічне представлення результатів кластеризації цим методом полягає в побудуванні залежностей величин ознак від номеру кластеру. Перевагами методу є простота, недоліками – невелика припустима кількість об'єктів, що кластеризуються та необхідність задання кількості кластерів. Критерієм припинення ітерацій є незмінність розподілу об'єктів за кластерами при здійсненні чергової ітерації.

Метод нечіткої кластеризації Fuzzy C-means. На відміну від чітких, алгоритми нечіткої кластеризації не відносять об'єкт однозначно до якого-небудь кластеру, а визначає для кожного кластеру ймовірність віднесення до нього відповідних об'єктів, формуючи, так звану, матрицю приналежності. Таким чином, на кожному кроці алгоритму кожен об'єкт одночасно відноситься до декількох кластерів.

Як і у попередніх алгоритмів вхідними даними для алгоритму нечіткої кластеризації Fuzzy C-means є матриця спостережень $\mathbf{X} = \{x_{ji}\}_{j=1, \dots, n}$ та кількість кластерів, що формується m .

Кластерна структура, що формується надається матрицею приналежності об'єктів до кластерів $\mathbf{M} = \{M_{kj}\}_{k=1, \dots, m, j=1, \dots, n}$. Тут M_{kj} – ступень належності j -го об'єкту k -му

кластеру, яка задовольняє наступним вимогам:

- $M_{kj} \in [0;1] \quad \forall k, j$;
- $\sum_{k=1}^m M_{kj} = 1 \quad \forall j$ – тобто кожен об'єкт повинен бути повністю розподілений між кластерами;
- $\sum_{j=1}^n M_{kj} \in (0; n) \quad \forall k$ – тобто жоден кластер не повинен бути порожнім або містити всі об'єкти.

Для оцінки якості розбиття використовується критерій розкиду, що показує суму відстаней від об'єктів до центрів кластерів з відповідними ступенями приналежності:

$$J = \sum_{k=1}^m \sum_{j=1}^n M_{kj}^w \cdot \rho(\bar{\bar{X}}_k, \bar{X}_j), \quad (2.6.13)$$

де $\rho(\bar{\bar{X}}_k, \bar{X}_j)$ – відстань між об'єктом $\bar{X}_j = (x_{j1}, \dots, x_{jp})$ та центроїдом k -го кластера $\bar{\bar{X}}_k = (x_{k1}, \dots, x_{kp})$, $w \in [0; +\infty)$ – експонентна вага, яка визначає нечіткість, розмитість кластерів. Чим більше це значення, тим значення матриці

приналежності для кожного об'єкту більш розмиті по кластерам і при $w \rightarrow +\infty$ її елементи приймають значення $M_{kj} = \frac{1}{m}$, тобто всі об'єкти з однаковим ступенем розподілені по всім кластерам. При $w=1$ алгоритм Fuzzy C-means вироджується в звичайний K-means. Теоретично обґрунтованого правила вибору ваги поки не існує, і зазвичай встановлюють $w=2$. Елементи матриці координат центроїдів кластерів $\bar{\mathbf{X}} = \{\bar{\mathbf{X}}_k\}_{k=1,\dots,m} = \{\bar{x}_{ki}\}_{i=1,\dots,p}^{k=1,\dots,m}$ визначаються як

$$\bar{x}_{ki} = \frac{\sum_{j=1}^n M_{kj}^w \cdot x_{ji}}{\sum_{j=1}^n M_{kj}^w} \quad (2.6.14)$$

Завданням нечіткої кластеризації є визначення значень матриці приналежності $\mathbf{M}$, яка мінімізує величину J , що визначається формулою (2.6.13). Значення елементів матриці приналежності об'єктів перераховуються за співвідношенням

$$M_{kj} = \begin{cases} \frac{1}{\rho_{kj}^{\frac{2}{w-1}} \sum_{l=1}^m \frac{1}{\rho_{lj}^{\frac{2}{w-1}}}} & \text{при } \rho_{kj} > 0 \\ 1 & \text{при } \rho_{kj} = 0 \end{cases} \quad (2.6.15)$$

Критерієм припинення ітерацій є умова малості зміни матриці приналежності при здійсненні чергової ітерації.

$$\|\mathbf{M} - \mathbf{M}^*\| = \sqrt{\sum_{k=1}^m \sum_{j=1}^n (M_{kj} - M_{kj}^*)^2} < \varepsilon. \quad (2.6.16)$$

Тут $\mathbf{M}$, $\mathbf{M}^*$ – матриці приналежності на поточному та попередньому кроках ітерації, ε – наперед визначене мале число.

Графічна схема алгоритму кластеризації за методом нечіткої кластеризації Fuzzy C-means представлена на рис. 2.6.3.

Недоліком алгоритму є високий ступінь залежності результуючого розбиття об'єктів на кластери від початкової матриці приналежності.

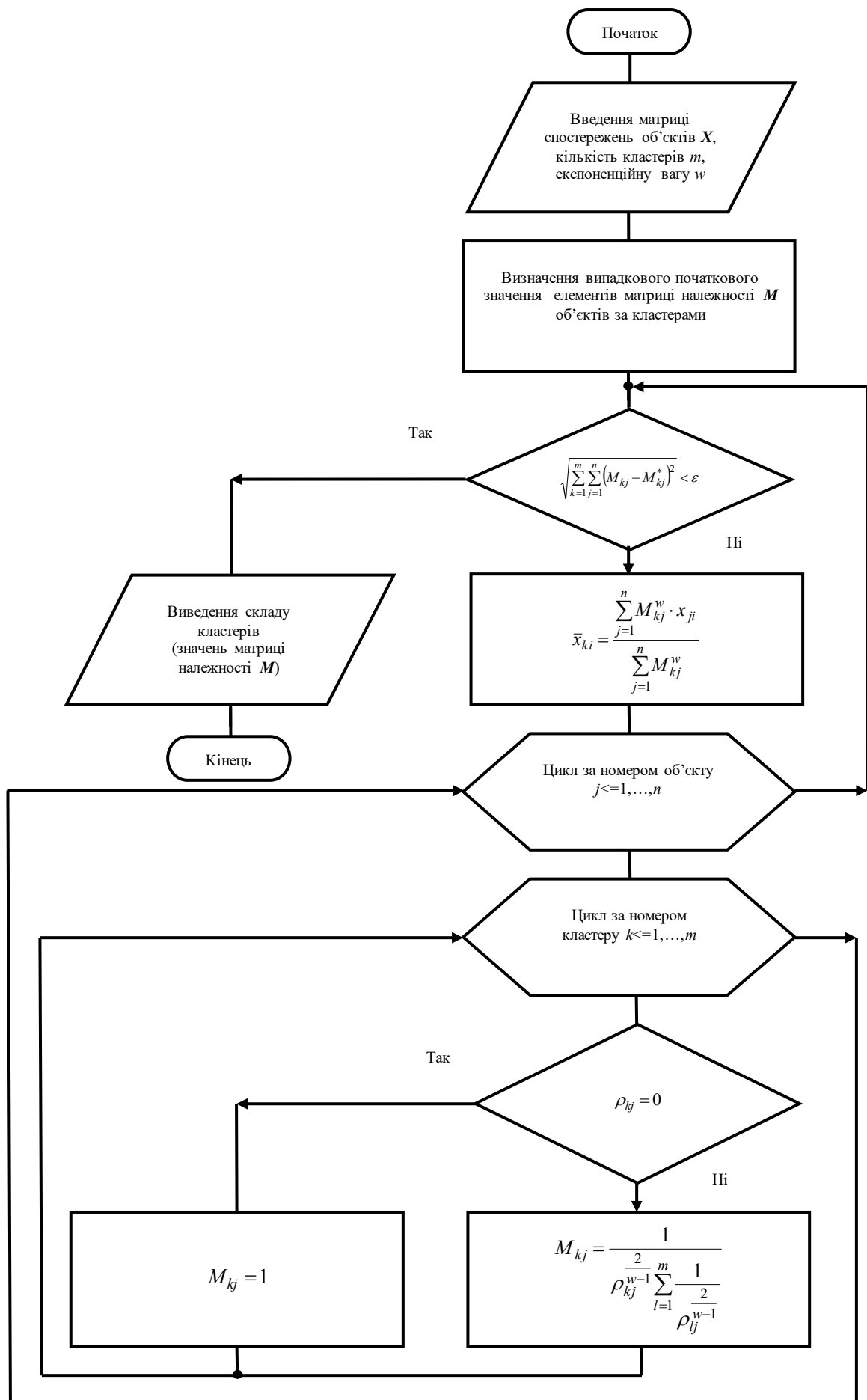


Рис. 2.6.3. Графічна схема алгоритму кластеризації за методом Fuzzy C-means

Постановка задачі

Дано

Багатовимірна вибірка результатів діагностики хворих з підозрою на апендицит. Рядки вибірки відповідають хворим, а стовбці – результативній ознаці – істинному діагнозу та факторним ознакам – симптомам хвороби. Ознаки вибірки надаються у порядковій шкалі та стандартизовані.

Потрібно

Сформувані: з вибірки даних, що відображають результати діагностики апендициту, підвибірку згідно номеру варіанту.

Провести: кластеризацію даних вибірки ієрархічним агломеративним методом та методами K-means, Fuzzy C-means за значеннями факторних ознак (симптомів хвороби) для встановлення необхідності оперативного втручання.

Визначити: відносні частоти виникнення помилок першого та другого роду при встановленні необхідності оперативного втручання.

Дослідити: залежність середньої дисперсії номеру кластеру від коефіцієнта експонентної ваги.

Послідовність виконання роботи

В табл. Д.6.1, що міститься у додатку 6 представлені варіанти, що відображають результати діагностики апендициту. Кожна варіанта (рядок) відповідає одному хворому. В першому стовпчику вказаний точний діагноз, який відображає ступінь тяжкості захворювання, (1 – непідтверджений діагноз, 2 – катаральний апендицит, 3 – флегмозний апендицит, 4 – гангренозний апендицит), в стовпчиках з другого по дев'ятий – значення симптомів x_1 – x_8 , які виражені в порядкових шкалах:

x_1 – болі в правій підвздошній області 1 – незначні, 2 – виражені;

x_2 – тривалість болей 1 – до 12 годин, 2 – 13 – 24 годин, 3 – 25 – 48 годин, 4 – понад 2 діб;

x_3 – частота пульсу 1 – до 80 уд/хв, 2 – 80 – 100 уд/хв, 3 – понад 100 уд/хв;

x_4 – лейкоцити крові 1 – до 8 тис., 2 – 8–14 тис., 3 – понад 14 тис.;

x_5 – зміни язика 1 – не обкладений, 2 – обкладений,;

x_6 – симптом Щоткіна – Блюмберга 1 – відсутній, 2 – виражений;

x_7 – симптом Розвіга 1 – відсутній, 2 – виражений;

x_8 – захисне м'язове напруження 1 – відсутнє, 2 – виражено.

Для проведення кластерного аналізу вони додатково переведені у рангову шкалу і стандартизовані та містяться у файлах *standard.txt*, *standard.xls*.

Відповідно до номеру варіанта, використовуючи середовище математичних розрахунків MathCAD та програмну оболонку інтерпретатора Python, наприклад Anaconda:

1. Імпортувати дані вибірки у середовище розрахунків та сформувати підвибірку з варіант з діапазоном індексів: $N_C + N_G - 1 \div N_C + N_G + 3$, $N_C + N_G + 23 \div N_C + N_G + 27$, $N_C + N_G + 49 \div N_C + N_G + 53$, $N_C + N_G + 74 \div N_C + N_G + 78$. Тут N_C – номер здобувача вищої освіти за журналом, N_G – номер групи.

2. Провести кластеризацію даних методом K-means для випадку 4-х кластерів. Код відповідної MathCAD-програми міститься у файлі *Kmeans.mcd*, а фрагмент програмного блоку – у додатку 7. Визначити коефіцієнти парної кореляції факторних ознак (симптомів хвороби) з результативною ознакою – істинним діагнозом. Визначити по мінімальному значенню симптому який найбільш корельований з діагнозом, номеру кластеру, що відповідає непідтвердженому діагнозу. Вважаючи, що нульова гіпотеза формулюється як “апендициту немає, хірургічне втручання не потрібне”, знайти відносну частота помилки першого роду – апендициту немає, але призначається втручання та відносну частоту помилки другого роду – апендицит є, але втручання не призначається.
3. Провести кластеризацію даних методом Fuzzy C-means для випадку 4-х кластерів і коефіцієнта експонентної ваги $w=1.5$. Код відповідної MathCAD-програми міститься у файлі *Cmeans.mcd*, а фрагмент програмного блоку – у додатку 7. Визначити коефіцієнти парної кореляції факторних ознак (симптомів хвороби) з результативною ознакою – істинним діагнозом. Визначити по мінімальному значенню симптому який найбільш корельований з діагнозом, номеру кластеру, що відповідає непідтвердженому діагнозу. Знайти відносні частоти помилок першого та другого роду. Побудувати графік залежності середньої дисперсії номеру кластеру від коефіцієнта експонентної ваги.
4. Провести кластеризацію даних ієрархічними агломеративними методами та побудувати дендрограму, на дендрограмі позначити горизонталь, що відповідає випадку 4-х кластерів. Група 1 використовує метод Уорда, група 2 – повного, група 3 – середнього зв'язку. Код Python-програми кластеризації даних ієрархічними агломеративними методами з побудованням дендрограми міститься у додатку 8. Експортувати дендрограму у вигляді графічного файлу і за допомогою графічного редактору різними кольорами виділити варіанти, що відповідають можливим значенням істинного діагнозу.

Вимоги до звіту

Звіт роботі повинен містити:

1. Назву дисципліни та лабораторної роботи.
2. Прізвище, ім'я та по батькові здобувача вищої освіти, шифр групи, номер варіанта.
3. Об'єкт, предмет і мету лабораторної роботи.
4. Коди програм, що реалізують поставлені завдання.
5. Сформовану підвбірку стандартизованих результатів діагностики апендициту.
6. Результати кластеризації даних методами K-means та Fuzzy C-means для випадку 4-х кластерів.
7. Результати розрахунку відносних частот помилок першого та другого роду для обох методів.

8. Графік залежності середньої дисперсії номеру кластеру від коефіцієнта експонентної ваги для методу Fuzzy C-means.
9. Дендрограму кластеризації даних ієрархічними агломеративними методами з горизонталлю, що відповідає випадку 4-х кластерів та кольоровим виділенням варіант, що відповідають можливим значенням істинного діагнозу.

10. Висновки.

Контрольні питання і завдання

1. Сформулюйте математичну постановку завдання кластеризації.
2. Наведіть властивості мір відстані та подібності.
3. Знайдіть манхеттенську відстань між векторами $\vec{X}_r = (1, 0, -1, 2)$; $\vec{X}_v = (3, 3, -2, -1)$.
4. Назвіть основні групи методів кластерного аналізу.
5. Чому дорівнює число варіантів кластеризації 4-х об'єктів на 3 кластери?
6. Для матриці спостережень $X = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}$ знайти матрицю евклідових відстаней та побудувати дендрограму методом ближнього зв'язку.
7. Назвіть властивості матриці приналежності об'єктів до кластерів.

2.7. Лабораторна робота № 7

Ознайомлення з методами прийняття рішень при класифікації даних за допомогою нейронних мереж

Об'єкт – нейронні мережі. Предмет – нейронні мережі персептронів, які призначені для класифікації даних. Мета – отримання навичок програмування найпростіших нейронних мереж для прийняття рішень при класифікації даних.

Стислі теоретичні відомості

Принцип роботи штучної нейронної мережі відповідає алгоритму роботи біологічних нейронів. Нейрон – обчислювальна одиниця, яка отримує інформацію і виробляє над цією інформацією будь-які обчислення. Нейрони є найпростішою структурною одиницею будь-якої нейронної мережі. Їх множини, упорядковані певним складають в шари, які, в свою чергу поєднуються в мережу в цілому. Всі нейрони функціонують приблизно однаковим чином, однак бувають деякі окремі випадки нейронів, що виконують специфічні функції. На рис. 2.7.1 представлений приклад топології нейронної мережі.

Розрізняють три основних типи нейронів: вхідні – нейрони шару, який одержує інформацію з зовні (синій колір), приховані – нейрони кількох шарів, що обробляють інформацію (червоний колір), вихідні – нейрони шару, що

представляє результати обчислень (зелений колір).

Крім самих нейронів складовою частиною мережі є синапси. Синапс – зв'язок, що з'єднує вихід одного нейрона з входом іншого. Під час проходження сигналу через синапс сигнал може посилюватися або слабшати. Параметром синапсу є вага (деякий коефіцієнт, може бути будь-яким дійсним числом), через який інформація, що передається від одного нейрона до іншого, може змінюватися. За допомогою зважування вхідна інформація проходить обробку, після чого отримується результат.

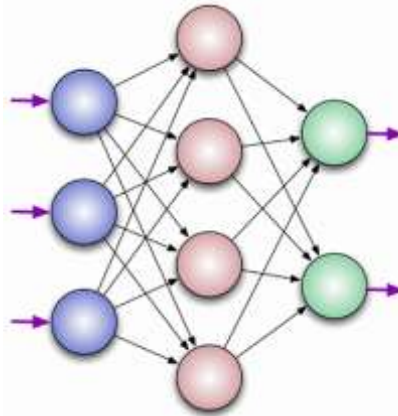


Рис. 2.7.1. Приклад топології нейронної мережі

На рис. 2.7.2 зображена графічна схема моделі штучного нейрона. На вхід подаються числа (сигнали) x_i , після вони множаться на ваги w_i (кожен сигнал на свою вагу) і підсумовуються. Функція активації вираховує остаточний сигнал y і подає його на вихід. Функція активації дозволяє проходити або не проходити сигналам від одного нейрону до іншого, або на вихід мережі в залежності від інформації, яку вони передають. Тобто якщо інформація є важливою, то функція пропускає її, а якщо інформації мало або вона недостовірною, то функція активації не дозволяє їй пройти далі.

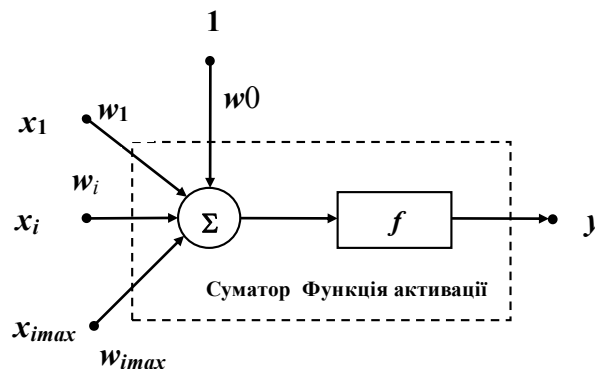


Рис. 2.7.2. Графічна схема моделі штучного нейрона.

У табл. 2.7.1 представлені деякі види активаційних функцій нейрона. Функція активації повинна бути однаковою для всіх нейронів всередині одного шару, однак для різних шарів можуть вибиратися різні функції активації. У

штучних нейронних мережах вибір функції обумовлюється областю використання і є важливим завданням.

Таблиця 2.7.1. Найбільш поширені функції активації штучного нейрона

Назва	Математичне подання
Лінійна	ax
Порогова (одинична, функція Хевісайда)	$\begin{cases} 0 & \text{при } x < x_0 \\ 1 & \text{при } x \geq x_0 \end{cases}$
Функція з лінійним порогом	$\begin{cases} 0 & \text{при } x < 0 \\ ax & \text{при } 0 \leq x \leq \frac{1}{a} \\ 1 & \text{при } x > \frac{1}{a} \end{cases}$
ReLU	$\begin{cases} 0 & \text{при } x < 0 \\ x & \text{при } x \geq 0 \end{cases}$
Leaky ReLU	$\begin{cases} ax & \text{при } x < 0, \ 0 < a \ll 1 \\ x & \text{при } x \geq 0 \end{cases}$
Сигмоїдна	$\frac{1}{1 + \exp(-x)}$
Гіперболічний тангенс	$\frac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)}$

Застосування нейронної мережі передбачає її попереднє навчання за даними навчальної вибірки. Схематично процес навчання представлений на рис.2.7.3.

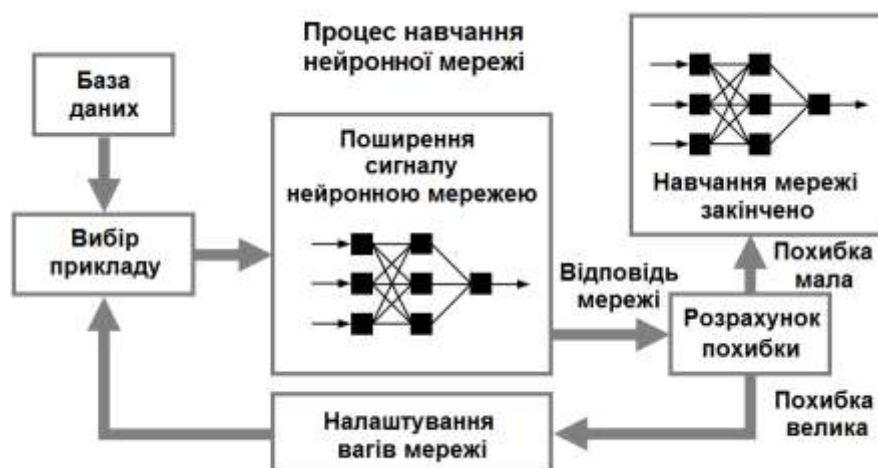


Рис. 2.7.3. Схема процесу навчання нейронної мережі

Нейронні мережі набувають все більшої популярності і область їх використання також розширюється. Список деяких областей, де застосовуються штучні нейронні мережі:

Введення і обробка інформації. Розпізнавання текстів, що подані у вигляді растрових графічних даних, розпізнавання голосових команд, голосове введення тексту.

Безпека. Розпізнавання облич і різних біометричних даних, аналіз трафіку в мережі, виявлення підробок.

Інтернет. Таргетингова реклама, капча, блокування спаму, асоціативний пошук інформації.

Зв'язок. Маршрутизація пакетів, стиснення відеоінформації, збільшення швидкості кодування і декодування інформації.

Персептрон є найпростішою моделлю нейромережі, що може складатися з одного нейрона (рис.2.7.2). Нейрон може мати довільну кількість входів, а вхідний сигнал одного з них зазвичай тотожно дорівнює 1. Цей одиничний вхід називають зсувом. Така проста мережа вирішувати прості завдання: виконувати класифікацію лінійно роздільних даних, інтерполювати часові ряди, що мають лінійний функціональний тренд. Для лінійно роздільних даних завжди можна провести пряму (або гіперплощину в багатовимірному просторі), що відокремлюють дані одного класу від даних іншого класу. Це і звуть лінійною роздільності об'єктів. Тоді модель мережі, що розглядається, зможе вирішити поставлене завдання класифікації. Якщо ж об'єкти лінійно нероздільні, то мережа з одного нейрона з ними не впорається. Для цього існують більш складні архітектури нейронних мереж що складаються з множини персептронів, об'єднаних в шари.

Розглянемо приклад застосування персептрону для класифікації лінійно роздільних даних. Дані представляють собою крапки числової площини з координатами (x_1, x_2) , які рівномірно розподілені в квадраті $x_1, x_2 \in (0,1)$. Навчальна вибірка окрім матриці вказаних вхідних даних сигналів містить вектор вихідних сигналів, які формуються за наступним правилом

$$Y_k = \begin{cases} 0 & \text{при } X_{k2} < aX_{k1} + b \\ 1 & \text{при } X_{k2} \geq aX_{k1} + b \end{cases}, \quad a, b \in (0,1), \quad (2.7.1)$$

тобто, якщо крапка знаходиться нижче прямої $x_2 = ax_1 + b$, вона відноситься до класу 0, інакше – до класу 1.

Далі будуть прийняті такі позначення $\{w_0^n\}$, $\{w_i^n\}_{i=1, \dots, imax}$ – ваги синапсів на n -му кроці навчання, $imax=2$ – кількість входів персептрону без входу зсуву, $\{X_{ki}\}_{k=1, \dots, kmax}^{i=1, \dots, imax}$ – навчальна вибірка вхідних сигналів, $kmax$ – об'єм вибірки, який може

бути довільним, $\{Y_k\}_{k=1, \dots, kmax}$ – навчальна вибірка вихідних сигналів. При навчанні мережі здійснюється її тренування на кожній варіанті навчальної вибірки впродовж

$nmax$ циклів. Фактичний вихідний сигнал персептрону на n -му циклі навчання при подачі на вхід вхідних сигналів k -ї варіанти навчальної вибірки при цьому складає

$$y_k^n = f\left(w_0 + \sum_{i=1}^{imax} w_i^n X_{ki}\right). \quad (2.7.2)$$

Функція активації f при цьому є пороговою, а вихідний сигнал – бінарним. Похибка мережі на n -му циклі навчання визначається як сума за всіма варіантами навчальної вибірки абсолютних значень різниць між очікуваним та фактичним значенням вихідного сигналу мережі

$$\varepsilon^n = \sum_{k=1}^{kmax} |Y_k - y_k^n|. \quad (2.7.3)$$

При цьому умовою припинення навчання може бути як досягнення необхідного значення похибки, так і просте обмеження кількості ітерацій певною величиною $nmax$. Правила налаштування ваг синапсів залежить від архітектури мережі. Для персептрону може бути використане співвідношення

$$\begin{cases} w_0^{n+1} = w_0^n + \sum_{k=1}^{kmax} \eta (Y_k - y_k^n) \\ w_i^{n+1} = w_i^n + \sum_{k=1}^{kmax} \eta (Y_k - y_k^n) X_{ki} \text{ при } 1 \leq i \leq imax \end{cases}, \quad (2.7.4)$$

де $\eta \in (0,1)$ – швидкість навчання. У якості початкових значень ваг синапсів w_i^0 беруться випадкові величини, які рівномірно розподілені на сегменті $(-0.1, 0.1)$. Наведене вище правило налаштування ваг синапсів носить назву дельта-правила і є узагальненням правил Хебба. Останні були сформульовані для персептрону, який працює з бінарними вхідними і вихідними сигналами і мають вигляд:

- якщо сигнал персептрону невірний, і дорівнює нулю, то необхідно збільшити ваги тих входів, на які була подана одиниця;
- якщо сигнал персептрону невірний і дорівнює одиниці, то необхідно зменшити вагу тих входів, на які була подана одиниця.

Дійсно, якщо вихідний сигнал бінарного класифікатора невірний, і дорівнює нулю $y_k^n = 0$, $Y_k - y_k^n = 1$ і, з урахуванням невід’ємності величини X_{ki} , застосування дельта-правила веде до збільшення ваг всіх синапсів, що викликає виправлення значення вихідного сигналу на наступному циклі навчання, $y_k^{n+1} = 1$, або принаймні збільшення сигналу на виході суматора, який є аргументом функції активації. Навпаки, якщо вихідний сигнал бінарного класифікатора невірний, і дорівнює одиниці $y_k^n = 1$, $Y_k - y_k^n = -1$, застосування дельта-правила веде до зменшення ваг всіх синапсів, що викликає виправлення значення вихідного сигналу на наступному

циклі навчання, $y_k^{n+1} = 0$, або принаймні збільшення сигналу на виході суматора.

Вихідний сигнал навченої мережі, який виникає при поданні на її входи сигналу $\{x_i\}_{i=1,\dots,imax}$ становить

$$y^{nmax} = f\left(w_0 + \sum_{i=1}^{imax} w_i^{nmax} x_i\right). \quad (2.7.5)$$

Зауважимо, що, оскільки умовою віднесення крапки, яка класифікується, до класу 1 є $x_2 \geq ax_1 + b$, а умовою того, що вихідний сигнал мережі дорівнюватиме 1 – $w_0^{nmax} + w_1^{nmax}x_1 + w_2^{nmax}x_2 \geq 0$, для синапсів навченої мережі виконуються умови

$$a \approx -\frac{w_1^{nmax}}{w_2^{nmax}}, \quad b \approx -\frac{w_0^{nmax}}{w_2^{nmax}}. \quad (2.7.6)$$

Програмна реалізація нейронної мережі бінарного класифікатора засобами системи MathCAD включає підготовчий фрагмент у якому задаються параметри мережі, формуються навчальна та тестова вибірки, визначаються початкові ваги синапсів, а також програмні блоки навчання та тестування мережі. Вхідними даними підготовчого блока відповідно виступають $a, b, \eta, kmax, mmax, nmax$. На його виході формуються значення $\{X_{ki}\}_{k=1,\dots,kmax, i=1,\dots,imax}$,

$\{Y_k\}_{k=1,\dots,kmax}$, $\{X_{tmi}\}_{m=1,\dots,mmax, i=1,\dots,imax}$, $\{Y_{tm}\}_{m=1,\dots,mmax}$, $\{w_0^0\}$, $\{w_i^0\}_{i=1,\dots,imax}$. На виході

програмного блока навчання отримуються значення $\{w_0^{nmax}\}$, $\{w_i^{nmax}\}_{i=1,\dots,imax}$ та ε^{nmax} , а на виході програмного блока тестування – значення $\{y_m^{nmax}\}_{m=1,\dots,mmax}$, які відповідають елементам тестової вибірки.

Мережа швидко навчається, для отримання нульової похибки потрібно близько десяти циклів навчання, при цьому ваги синапсів збігаються до значень, які задовольняють співвідношенням (2.7.6), що свідчить про правильність функціонування мережі (рис. 2.7.4).

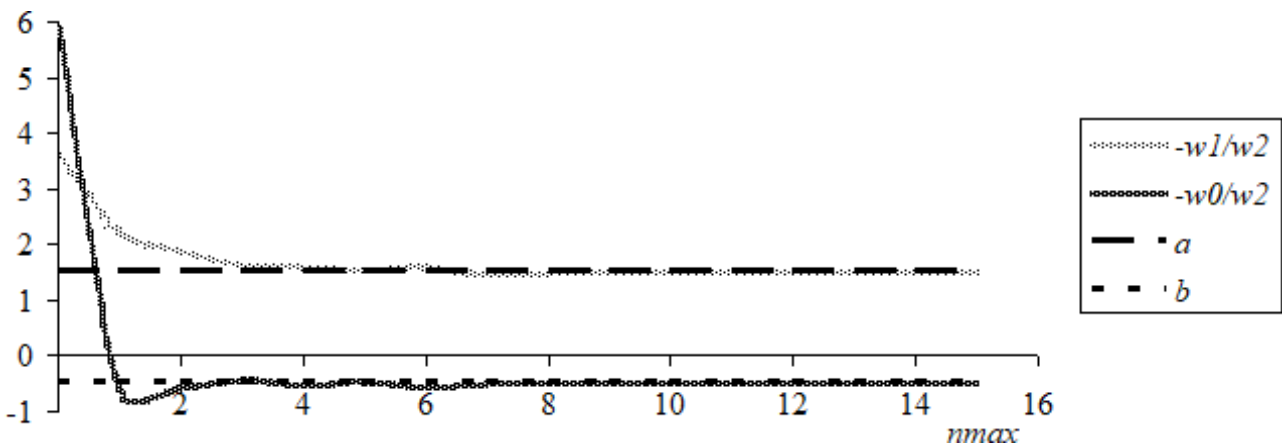


Рис. 2.7.4. Залежності ваг синапсів від кількості циклів навчання

Багатошаровий перцептрон є узагальненням одношарового перцептрона і складається з вхідного шару, одного чи декількох прихованих шарів обчислювальних нейронів і вихідного шару. Топологія перцептрону з одним прихованим шаром представлена на рис. 2.7.1.

Багатошаровий перцептрон також зветься нейронною мережею прямого поширення, використовується абревіатура MLP (Multiple Layer Perceptron).

Якщо записати вихідний сигнал навченої мережі, яка складається з одного шару, що містить довільну кількість нейронів у векторному вигляді

$$\vec{y} = f(\mathbf{w}\vec{x} + \vec{w}0), \quad (2.7.7)$$

для вихідного сигналу $\vec{y}_n$ мережі, яка складається з довільної кількості n таких шарів, можна отримати рекурентне співвідношення

$$\vec{y}_n = f_n(\mathbf{w}_n\vec{y}_{n-1} + \vec{w}0_n), \quad (2.7.8)$$

де f_n – функція активації n -го (вихідного) шару, $\vec{y}_{n-1}$ – вихідний сигнал останнього прихованого шару (вхідний сигнал вихідного шару), $\mathbf{w}_n$, $\vec{w}0_n$ – матриця ваг синапсів інформаційних входів і вектор ваг синапсів входів зсувів n -го (вихідного) шару.

У додатку 9 наведений програмний код мовою Python, який: здійснює генерацію набору даних (крапок на площині), що характеризується двома ознаками та може бути розбитий на два класи, при цьому ознаками є координати крапок, класи представляють собою множини крапок у вигляді двох напівмісяців з розмитими межами; будує і навчає нейронну мережу у вигляді багатошарового перцептрону з одним прихованим шаром; візуалізує межі розділу класів, що побудовані на основі навчальної вибірки і крапок даних навчальної (або контрольної) вибірки.

На рис. 2.7.5 а) представлений результат роботи програми з візуалізацією межі розділу класів, що побудовані на основі навчальної вибірки і крапок даних навчальної вибірки, а на рис. 2.7.5 б) – крапок даних контрольної вибірки.

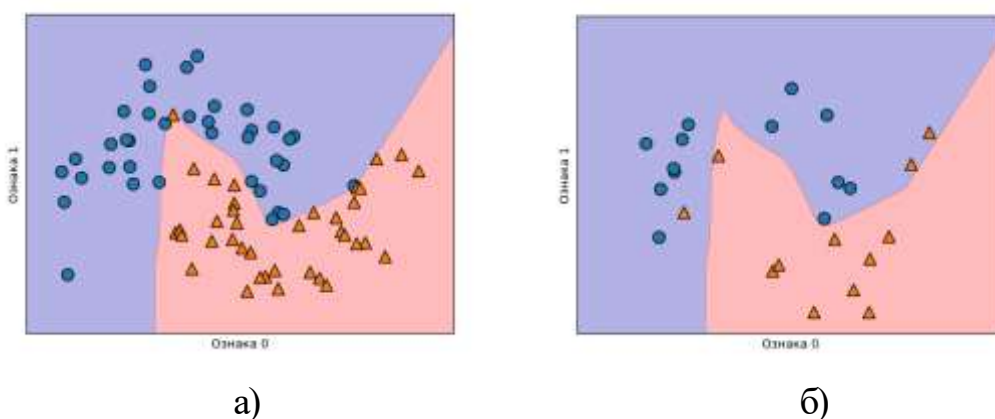
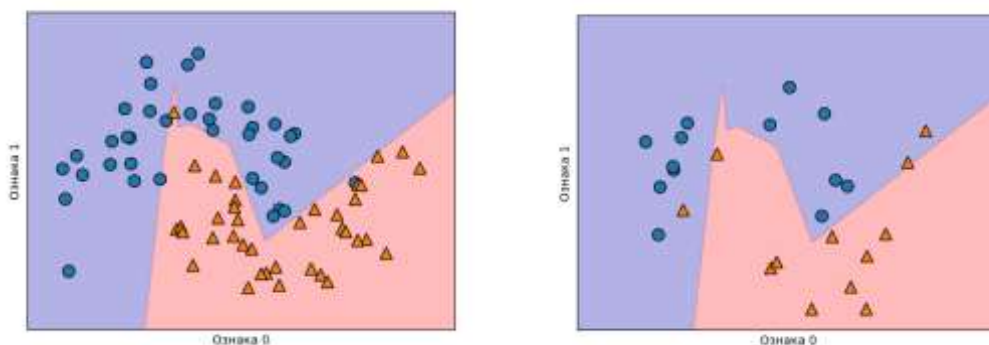


Рис. 2.7.5. Крапки вхідних даних, що класифікуються, межа розділу класів, яку будує нейромережа з кількістю нейронів прихованого шару 100 і функцією активації ReLU

З них можна бачити, що по-перше межа розділу класів досить якісно дискримінує дані різних класів для обох вибірок, а по-друге, що нейронна мережа будує нелінійну і досить гладку межу розділу. Насправді межа є кусково лінійною і гладкість пояснюється значною кількістю прихованих вузлів мережі (нейронів), оскільки за замовчуванням MLP використовує 100 прихованих вузлів. При зменшенні їх числа їх числа до 10-ти якість дискримінації суттєво не погіршується, але проявляється кускова лінійність межі, оскільки тепер вона складається з 10-ти прямолінійних відрізків (рис. 2.7.6 а, б).

Зміни в програмному коді:

```
mlp = MLPClassifier(solver='lbfgs', max_iter=100000, random_state=0,
hidden_layer_sizes=[10])
```



а)

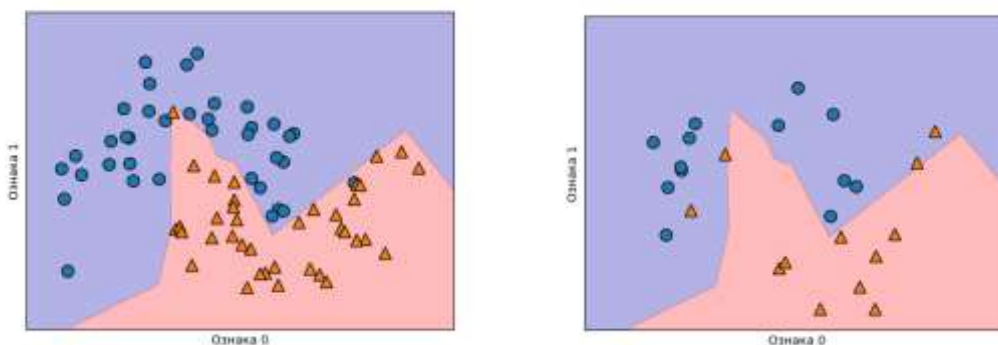
б)

Рис. 2.7.6. Крапки вхідних даних, що класифікуються, межа розділу класів, яку будує нейромережа з кількістю нейронів прихованого шару 10 і функцією активації ReLU

Якщо необхідно отримати більш гладку межу розділу класів, замість збільшення кількості прихованих вузлів, можна додати другий прихований шар (рис. 2.7.7 а, б).

Зміни в програмному коді:

```
mlp = MLPClassifier(solver='lbfgs', max_iter=100000, random_state=0,
hidden_layer_sizes=[10, 10])
```



а)

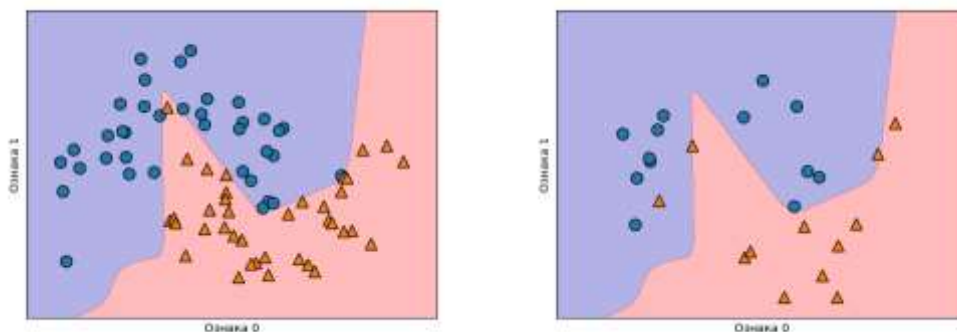
б)

Рис. 2.7.7. Крапки вхідних даних, що класифікуються, межа розділу класів, яку будує нейромережа з двома прихованими шарами з кількістю нейронів 10,10 і функцією активації ReLU

У якості альтернативного варіанту можна використовувати більш гладку функцію активації, наприклад гіперболічний тангенс (рис. 2.7.8 а, б).

Зміни в програмному коді:

```
mlp = MLPClassifier(solver='lbfgs',max_iter=100000, activation='tanh',
random_state=0,hidden_layer_sizes=[10])
mlp.fit(X_train, y_train)
```



а)

б)

Рис. 2.7.8. Крапки вхідних даних, що класифікуються, межа розділу класів, яку будує неймережа з кількістю нейронів прихованого шару 10 і функцією активації гіперболічний тангенс

Одним із завдань лабораторної роботи є створення нейронної мережі багат шарового персептрону, яка реалізує бінарну класифікацію випадкових даних. При цьому імовірнісний розподіл даних дещо відрізняється від розглянутого вище. Дані представляють собою крапки площини, які рівномірно розподілені у прямокутній області $[x_{\min}; x_{\max}] \cdot [y_{\min}; y_{\max}]$, в яку вписаний графік функції $y=f(x)$. Межі незалежної змінної $[x_{\min}; x_{\max}]$ та вид функції залежать від варіанту завдання. Належність крапки до певного класу (0 або 1) визначається її розташуванням нижче, або вище графіку функції.

Для розіграшу координат точок, які рівномірно розподілені у прямокутній області, може бути використана відома формула

$$x = x_{\min} + (x_{\max} - x_{\min}) \cdot \xi_1; \quad y = y_{\min} + (y_{\max} - y_{\min}) \cdot \xi_2, \quad (2.7.9)$$

де ξ_1, ξ_2 – незалежні псевдовипадкові числа, які рівномірно розташовані на сегменті $[0,1]$.

Для розіграшу координат точок, які рівномірно розподілені у круговій області радіусом R з центром в точці (x_0, y_0) використовується співвідношення

$$x = x_0 + \sqrt{\xi_1} \cdot R \cdot \cos(2 \cdot \pi \cdot \xi_2); \quad y = y_0 + \sqrt{\xi_2} \cdot R \cdot \sin(2 \cdot \pi \cdot \xi_2). \quad (2.7.10)$$

Рис. 2.7.9 ілюструє помилкову класифікацію об'єктів за синусоїдною функцією, що визначає істинні межі класів. Невірно класифіковані об'єкти показані зеленим кольором.

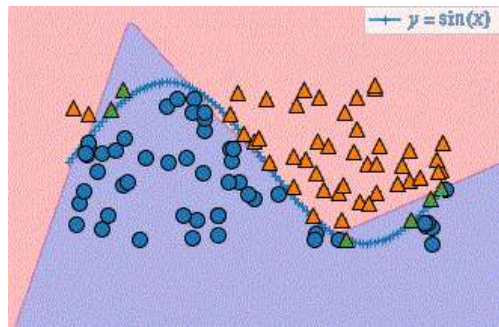


Рис. 2.7.9. Крапки вхідних даних, що класифікуються, межа розділу класів, яку буде нейромережа з кількістю нейронів прихованого шару 2 і функцією активації ReLU та графік дискримінуючої функції $y = \sin(x)$ з виділенням помилкових класифікацій, які показані зеленим кольором.

Дано

Середовище програмування мовою Python, дискримінуюча функція, яка визначає приналежність до певного класу даних, які угруповуються та межі варіації її незалежної змінної.

Потрібно

Створити Python-програму, що реалізує нейронну мережу у вигляді трьохшарового персептрону, який призначений для бінарної класифікації даних згідно варіанту завдання.

Рекомендації щодо виконання роботи

1. Передбачається, що здобувач вищої освіти має засоби компіляції мови Python, які при відсутності треба інсталиювати.

2. Згідно варіанту завдання запрограмувати нейронну мережу багатошарового персептрону, яка реалізує бінарну класифікацію даних. Дані представляють собою крапки площини, які рівномірно розподілені у прямокутній області $[x_{\min}; x_{\max}] * [y_{\min}; y_{\max}]$, де $[y_{\min}; y_{\max}]$ – межі зміни дискримінуючої функції $y = f(x)$ на сегменті $[x_{\min}; x_{\max}]$ згідно варіанту завдання. Належність крапки до певного класу (0 або 1) визначається її розташуванням нижче, або вище графіку дискримінуючої функції. Сумарний об'єм навчальної і контрольної вибірок для всіх іспитів прийняти таким, що дорівнює 100. Використовувати один прихований шар, кількість його нейронів прийняти: 2, 5, 10. Скористуватися наступними функціями активації: 'identity', 'logistic', 'tanh', 'relu' (остання за замовченням).

3. Дослідити якість роботи нейронної мережі. Скласти таблицю з заголовком згідно табл. 2.7.2.

Таблиця 2.7.2. Заголовок таблиці для внесення результатів досліджень

Кількість нейронів прихованого шару	Функція активації	Кількість помилкових класифікацій
-------------------------------------	-------------------	-----------------------------------

4. Побудувати в єдиному полі сімейство графіків залежностей кількості помилкових класифікацій від кількості нейронів прихованого шару для різних функцій активації.

Вимоги до звіту

Звіт по роботі повинен містити:

1. Назву дисципліни та лабораторної роботи.
2. Прізвище, ім'я та по батькові здобувача вищої освіти, код групи.
3. Номер варіанту завдання.
4. Код програми, що реалізує поставлені завдання.
5. Знімки зображень екрану, які виникають в процесі чисельного експерименту.
6. Таблицю і сімейство графіків, які ілюструють якість роботи нейронної мережі.

Контрольні питання

1. Назвіть приклади галузей застосування нейронних мереж.
2. Назвіть типові функції активації нейронних мереж.
3. Вербально опишіть основні кроки навчання нейронної мережі.
4. Сформулюйте правила Хебба. Для якого виду персептрону вони використовуються?
5. Запишіть співвідношення, яке може бути використане для обчислення похибки мережі на певному циклі навчання.
6. Поясніть сенс параметру `random_state` функції `make_moons`. В яких випадках його доцільно використовувати?
7. Запишіть рекурентне співвідношення для вихідного сигналу багаточарового персептрону.

Варіанти завдань

Таблиця 2.7.3. Дискримінуюча функція та межі варіації її незалежної змінної

№	Функція $y=f(x)$	Межі незалежної змінної $[x_{\min}; x_{\max}]$
1	$\operatorname{atan}(x)$	$[-2; 2]$
2	$\operatorname{sh}(x)$	$[-2; 2]$
3	$\operatorname{ch}(x)$	$[-2; 2]$
4	$\sin(x)$	$[-\pi; \pi]$
5	$ x $	$[-1; 1]$
6	$\ln(x)$	$[1; 10]$
7	$1 - x^2$	$[-1; 1]$
8	$\cos(x^2)$	$[-1; 1]$
9	$\sqrt{x}$	$[0; 1]$
10	$x \cdot \sin(x)$	$[0; 2\pi]$
11	$\cos(x)^2$	$[0; \pi]$
12	$\sqrt[3]{x}$	$[0; 8]$
13	$\exp(-x) \cdot \sin(x)$	$[0; \pi]$

14	$\cot(x)$	$\left[\frac{\pi}{2}; 3\right]$
15	$\frac{1}{\sin(x)}$	$\left[\frac{\pi}{2}; 3\right]$
16	$\frac{1}{1-x}$	$[-1; 0,5]$
17	$\frac{1}{x}$	$[0,5; 2]$
18	$\frac{x-1}{x+1}$	$[0; 1,5]$
19	$\cos(x)$	$[0; 2\pi]$
20	$\frac{\sin(x)}{x^2}$	$[0,5; 2\pi]$
21	$\sin\left(\frac{1}{x}\right)$	$\left[0,3; \frac{\pi}{2}\right]$
22	$ \sin(x) $	$[-\pi; \pi]$
23	x^2	$[0; 2]$
24	$\cos(x) + \sin(x)$	$\left[0; \frac{\pi}{2}\right]$
25	$\sqrt{\sin(x)}$	$\left[0; \frac{\pi}{2}\right]$
26	$\exp\left(\frac{1}{x}\right)$	$[0,5; 2]$
27	$\exp(-x^2)$	$[-1; 1]$
28	x^3	$[-1; 1]$
29	$x \cdot \exp(-x)$	$[0,5; 2]$
30	$ x \cdot \sin(x)$	$[-\pi; \pi]$
31	$\tan(x)$	$[-1; 1]$
32	$\sin(x)$	$[-\pi; \pi]$
33	$\ln\left(\frac{1}{x}\right)$	$[0,1; 5]$
34	$\frac{\exp(-x)}{x}$	$[0,2; 2]$
35	$\exp(x)$	$[0; 1,5]$
36	$\sqrt{1-x^2}$	$[-1; 1]$
37	x^x	$[0; 1,5]$

38	$\sqrt{x} \cdot \exp(-x)$	[0;1]
39	$\exp\left(-\frac{1}{x}\right)$	[0,1;0,3]
40	$\frac{x+1}{x-1}$	[0,1;0,6]

3. КРИТЕРІЇ ОЦІНЮВАННЯ

Навчальні досягнення здобувачів вищої освіти за результатами вивчення курсу оцінюватимуться за шкалою табл. 3.1.

Таблиця 3.1. Шкала оцінювання навчальних досягнень здобувачів вищої освіти

Рейтингова шкала	Інституційна шкала
90–100	відмінно
74–89	добре
60–73	задовільно
0–59	незадовільно

Здобувач вищої освіти може отримати підсумкову оцінку з навчальної дисципліни на підставі оцінювання знань та практичних навичок за умови, якщо набрана сумарна кількість балів зі складових оцінювання становитиме не менше 60 балів.

Успішність складається з оцінок за теоретичну частину курсу, лабораторні роботи та самостійної роботи здобувача вищої освіти з підготовки до лабораторних робіт. Отримані бали додаються один до одного і є підсумковою оцінкою за вивчення навчальної дисципліни. Максимально здобувач може набрати 100 балів.

Максимальне оцінювання навчальних досягнень здобувачів вищої освіти за видами занять наведено в табл. 3.2.

Таблиця 3.2. Максимальне оцінювання навчальних досягнень здобувачів вищої освіти за видами занять

Теоретична частина	Лабораторні роботи	Самостійна робота з підготовки до лабораторних робіт	Разом
37	56	7	100

Теоретична частина оцінюється за результатами online тесту, який містить 37 запитань.

Лабораторні роботи приймаються за звітами і контрольними запитаннями до кожної з роботи.

Критерії оцінювання теоретичної частини курсу

Для кожного запитання тесту потрібно вибрати єдину правильну відповідь. Кількість варіантів відповіді в залежності від характеру запитання може складати від 3-х до 10-ти. Кожне з 37-ми питань оцінюється в 1 бал, який

нараховуються за правильну відповідь. У разі неправильної відповіді на запитання, здобувач отримує за нього 0 балів.

Критерії оцінювання лабораторних робіт

За кожну лабораторну роботу здобувач може отримати до 8 балів (усього 56 балів), а саме:

8 балів: програма чисельних розрахунків правильно функціонує, супроводжується достатньою кількістю коментарів, звіт з роботи оформлений згідно з методичними рекомендаціями і містить повні, інформативні та обґрунтовані висновки.

5 балів: програма чисельних розрахунків правильно функціонує, але містить незначні помилки, які суттєво не впливають на отримувані результати, або кількість коментарів недостатня, або звіт з роботи оформлений із незначними відхиленнями від методичних рекомендацій або висновки не повністю задовольняють вимогам повноти, інформативності та обґрунтованості.

2 бали: програма чисельних розрахунків функціонує, але містить помилки, які суттєво впливають на отримувані результати, або коментарі відсутні, або звіт з роботи оформлений із суттєвими відхиленнями від методичних рекомендацій, або висновки не задовольняють вимогам повноти, інформативності та обґрунтованості.

0 балів: програма чисельних розрахунків не функціонує, або звіт з роботи відсутній, або не містить висновків, або висновки протирічать фактично отриманим результатам.

У разі отримання позитивної оцінки здобувач вищої освіти може підвищити її шляхом очної співбесіди з викладачем. При цьому з кожної лабораторної роботи здобувач вищої освіти отримує 3 запитання з переліку контрольних запитань. За кожну правильну відповідь на запропоновані запитання нараховується 1 бал.

Критерії оцінювання самостійної роботи

За підготовку вхідних даних згідно з варіантом завдання до кожної лабораторної роботи здобувач отримує:

1 бал (усього до 7 балів).

0 балів: підготовку вхідних даних не виконано, дані розраховані невірно або не відповідають варіанту завдання.

СПИСОК ВИКОРИСТАНОЇ ТА РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ

1. Коряшкіна Л.С. Моделі й методи прийняття рішень: навч. посіб. / С.А. Ус, Л.С. Коряшкіна // Дніпро: НТУ «ДП», 2018. – 300 с.
2. Анікін В.К. Теорія прийняття рішень: конспект лекцій [Електронний ресурс] : навч. посіб. для студ. спец. 126 «Інформаційні системи та технології»/ В. К. Анікін, Є.В. Крилов, В.П. Пасько – Київ: КПІ ім. Ігоря Сікорського, 2023, 134 с.
3. Ус С.А. Прийняття рішень в умовах ризику. Методичні рекомендації до виконання лабораторних робіт з дисципліни «Теорія прийняття рішень» студентами напряму підготовки 6.040303 Системний аналіз / С.А. Ус // Дніпро : НГУ, 2014. – 35 с.
4. Катренко А.В. Теорія прийняття рішень / А.В. Катренко, В.А. Пасічник, В.П. Пасько. – Київ: Видавнича група ВНУ, 2009. – 448 с.
5. Волошин О. Ф. Моделі та методи прийняття рішень : навч. посіб. для студ. вищ. навч. закл. / О. Ф. Волошин, С. О. Мащенко. – 2-ге вид., перероб. та допов. – Київ: Видавничо-поліграфічний центр "Київський університет", 2010. – 336 с.
6. Бутко М.П. Теорія прийняття рішень. Підручник / М. П. Бутко, І. М. Бутко, В. П. Мащенко; Київ: «Центр учбової літератури», 2015. – 360 с.
7. Кушлик-Дивульська О.І. Основи теорії прийняття рішень. Навчальний посібник/ О.І Кушлик-Дивульська., Б.Р. Кушлик. – Київ, 2014. – 94 с.
8. Желдак Т.А. Нечіткі множини в системах управління та прийняття рішень: навч. посіб./ Т.А. Желдак, Л.С. Коряшкіна, С.А. Ус // Дніпро: НТУ «ДП», 2020. – 387 с.
9. Величко О.М. Групове експертне оцінювання та компетентність експертів/ О. М. Величко, Л. В. Коломієць, Т. Б. Гордієнко; за загал. ред. д-ра техн. наук О. М. Величка. – Одеса: ФОП Бондаренко М.О., 2015. – 286 с.
10. Солодка Н.О. Методичні вказівки до лабораторних занять з дисципліни «Теорія ігор» за освітнім рівнем «Магістр» для студентів спеціальності 122 «Комп'ютерні науки» / Н.О. Солодка. – Дніпро: ДВНЗ УДХТУ, 2023. – 24 с.
11. Кожевников А.В. Створення відкритої нейронної мережі бінарного класифікатора засобами системи Mathcad/ А.В. Кожевников, І.М. Удовик // Матеріали XVI міжнародної конференції "Проблеми використання інформаційних технологій у сфері освіти, науки та промисловості" – Дніпро: НТУ «Дніпровська політехніка», 2021. – С. 27–32.
12. Кожевников А.В. Створення відкритої нейронної мережі предиктора лінійного часового ряду засобами системи Mathcad/ А.В. Кожевников, І.М. Удовик // Матеріали XVI міжнародної конференції "Проблеми використання інформаційних технологій у сфері освіти, науки та промисловості" – Дніпро: НТУ «Дніпровська політехніка», 2021. – С. 45–49.

Приклад розв'язання завдання лінійного програмування

Знайти невід'ємні значення змінних y_j ($j=1,2$) – елементів рішення, які задовольняють обмеженням у вигляді системи нерівностей

$$\begin{cases} y_1 \geq 1, & y_1 \leq 2, \\ y_2 \geq 1, & y_2 \leq 2. \end{cases} \quad (\text{Д.1.1})$$

При цьому максимізується цільова функція

$$T(y_1, y_2) = y_1 + y_2. \quad (\text{Д.1.2})$$

Далі на рис. Д.1.1 приведені графічне рішення прямої задачі, що містить: графіки всіх лінійних обмежень, виділену область припустимих рішень, положення опорної прямої цільової функції і прямої цільової функції, яка перетинає точку оптимального рішення, точку (або множину точок) оптимального рішення, відкладені по осях графічного поля оптимальні значення елементів рішення – потужностей ГЕС та ТЕС.

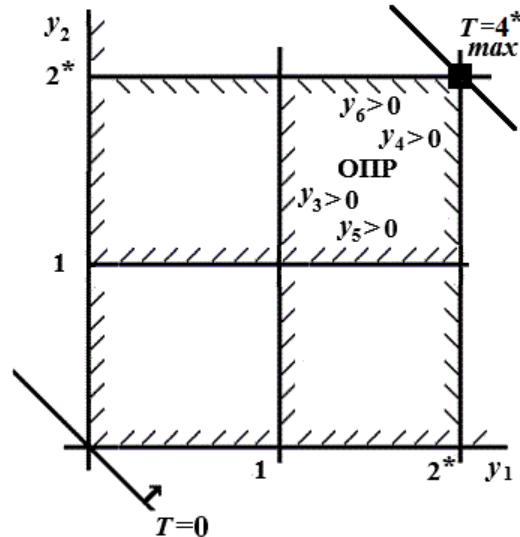


Рис. Д.1.1. Графічне рішення прямої задачі лінійного програмування

З рис. Д.1.1 випливає, що в точці оптимуму значення елементів рішення складають $y_1=2, y_2=2, L=4$.

Система обмежень (Д.1.1) у вигляді системи рівнянь виглядає як

$$\begin{cases} y_1 - y_3 = 1, & y_1 + y_4 = 2, \\ y_2 - y_5 = 1, & y_2 + y_6 = 2, \end{cases} \quad (\text{Д.1.3})$$

де y_j ($j=3..6$) є невід’ємними.

Система обмежень (Д.1.3) і цільова функція, які приведені до виглядів, що є зручними для складання симплекс-таблиці, записуються наступним чином

$$\begin{cases} y_3 = -1 - (-y_1), & y_4 = 2 - y_1, \\ y_5 = -1 - (-y_2), & y_6 = 2 - y_2. \end{cases} \quad (\text{Д.1.4})$$

$$T(y_1, y_2) = -(-y_1 - y_2). \quad (\text{Д.1.5})$$

Відповідна до (Д.1.4), (Д.1.5) симплекс-таблиця виглядає як

$$\begin{pmatrix} -1 & -1 & 0 \\ 2 & 1 & 0 \\ -1 & 0 & -1 \\ 2 & 0 & 1 \\ 0 & -1 & -1 \end{pmatrix}. \quad (\text{Д.1.6})$$

При цьому змінні y_j ($j=1,2$) є вільними, а y_j ($j=3..6$) – базисними.

Далі на рис. Д.1.2 – Д.1.6 наведені знімки зображень екрану, які виникають при знаходженні опорного та оптимального рішень прямої задачі, що включають поточні вигляди симплекс-таблиць та вікна зі значеннями розрішаючих рядків та стовбців на кожному кроці максимізації цільової функції. Рішення здійснюється за допомогою авторського програмного забезпечення. Індксація рядків і стовбців симплекс-таблиці здійснюється з нуля.

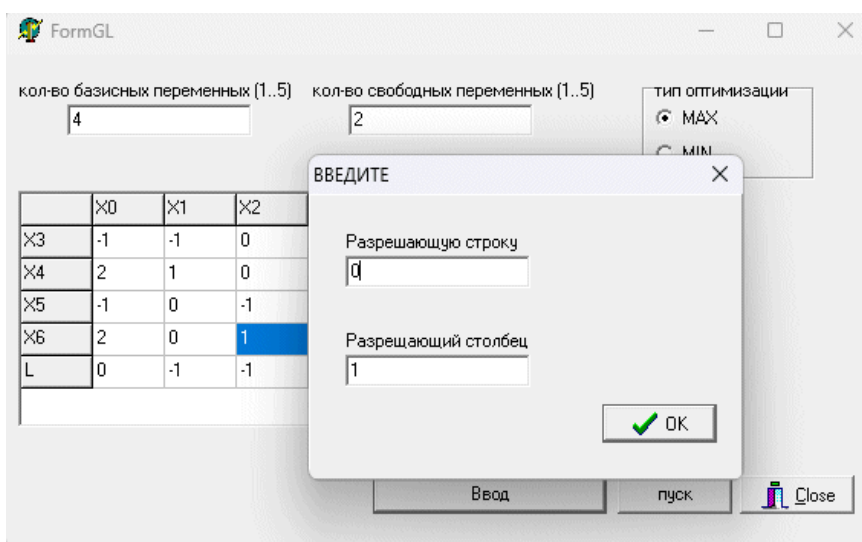


Рис. Д.1.2. Знімок зображення екрану, який виникає на першому кроці при знаходженні опорного рішення

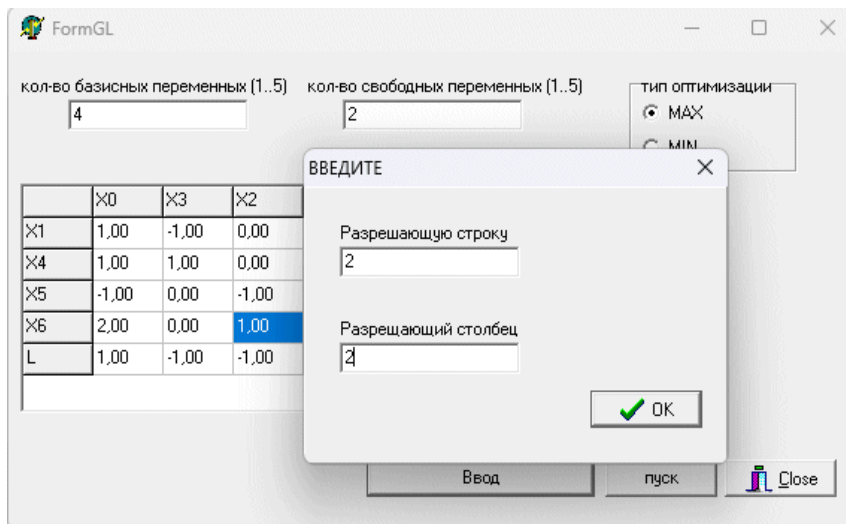


Рис. Д.1.3. Знімок зображення екрану, який виникає на другому кроці при знаходженні опорного рішення

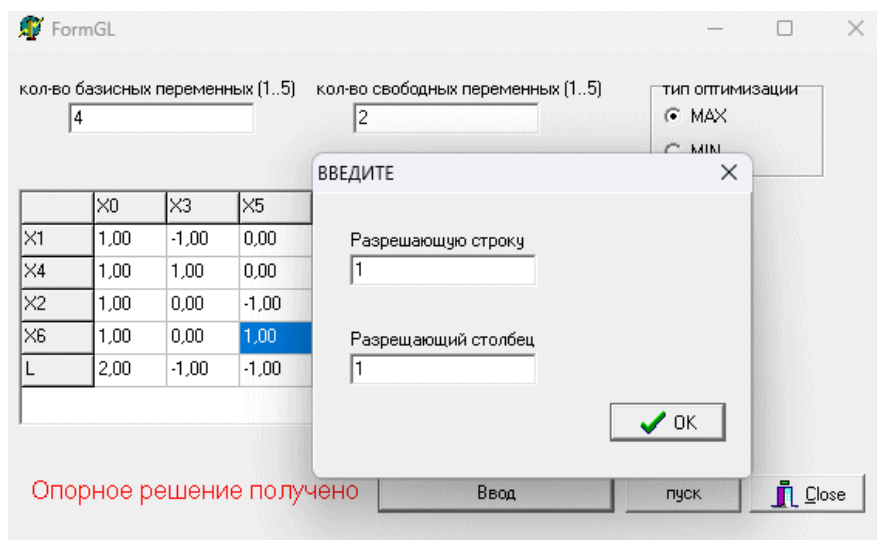


Рис. Д.1.4. Знімок зображення екрану, який виникає на першому кроці при знаходженні оптимального рішення

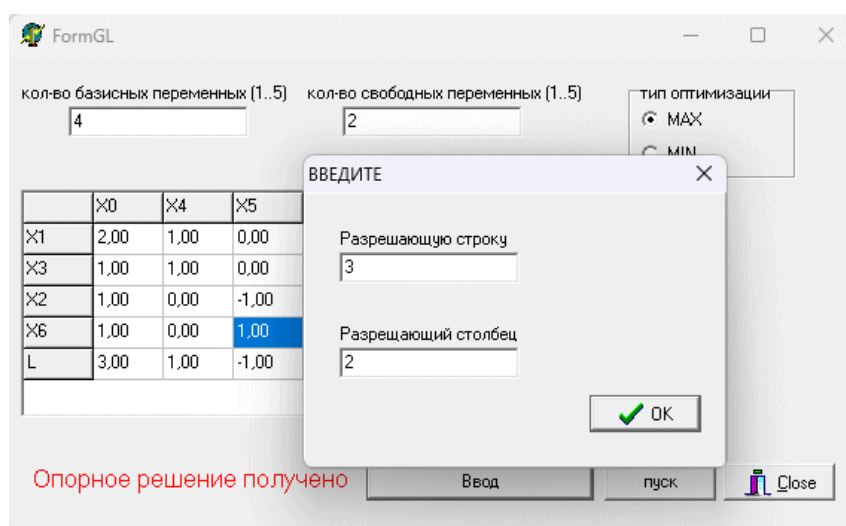


Рис. Д.1.5. Знімок зображення екрану, який виникає на другому кроці при знаходженні оптимального рішення

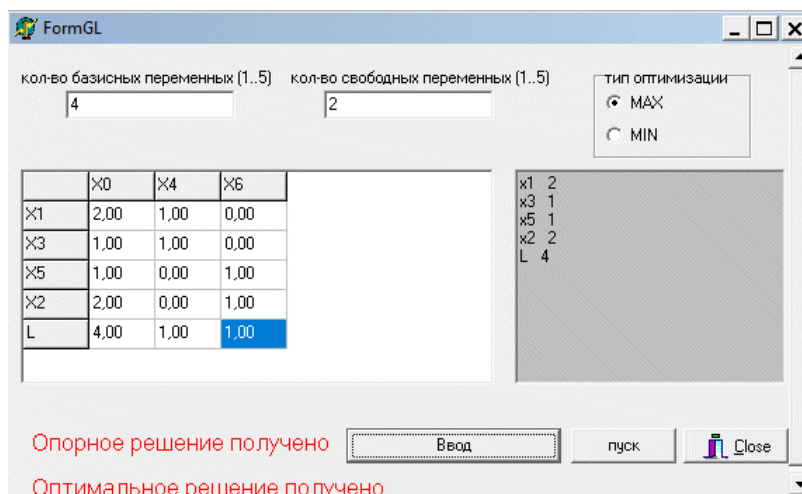


Рис. Д.1.6. Знімок зображення екрану, який відповідає оптимальному рішенню

З рис. Д.1.6 випливає, що в точці оптимуму значення елементів рішення складають $y_1=2, y_2=2, L=4$ і збігаються з отриманими графічним методом.

Двоїсте завдання лінійного програмування щодо розглянутого вище формулюється наступним чином. Знайти невід'ємні значення змінних x_i ($i=1..4$) які задовольняють обмеженням у вигляді системи нерівностей

$$\begin{cases} -x_1 + x_2 \geq 1, \\ -x_3 + x_4 \geq 1. \end{cases} \quad (\text{Д.1.7})$$

При цьому мінімізується цільова функція

$$L(x_1, x_2, x_3, x_4) = -x_1 + 2x_2 - x_3 + 2x_4. \quad (\text{Д.1.8})$$

Система обмежень (Д.1.7) у вигляді системи рівнянь виглядає як

$$\begin{cases} -x_1 + x_2 - x_5 = 1, \\ -x_3 + x_4 - x_6 = 1, \end{cases} \quad (\text{Д.1.9})$$

де y_j ($j=3..6$) є невід'ємними.

Система обмежень (Д.1.3) і цільова функція, які приведені до виглядів, що є зручними для складання симплекс-таблиці, записуються наступним чином

$$\begin{cases} -x_1 + x_2 - x_5 = 1, \\ -x_3 + x_4 - x_6 = 1. \end{cases} \quad (\text{Д.1.10})$$

$$T(x_1, x_2, x_3, x_4) = 2x_2 + 2x_4 - (-x_1 - x_3). \quad (\text{Д.1.11})$$

Відповідна до (Д.1.10), (Д.1.11) симплекс-таблиця виглядає як

$$\begin{pmatrix} -1 & 1 & -1 & 0 & 0 \\ -1 & 0 & 0 & 1 & -1 \\ 0 & 1 & -2 & 1 & -2 \end{pmatrix}. \quad (\text{Д.1.12})$$

При цьому змінні x_i ($i=1..4$) є вільними, а x_i ($i=5,6$) – базисними.

Далі на рис. Д.1.7 – Д.1.12 наведені знімки зображень екрану, які виникають при знаходженні опорного та оптимального рішень двоїстої задачі, що включають поточні вигляди симплекс-таблиць та вікна зі значеннями розрешаючих рядків та стовбців на кожному кроці максимізації цільової функції.

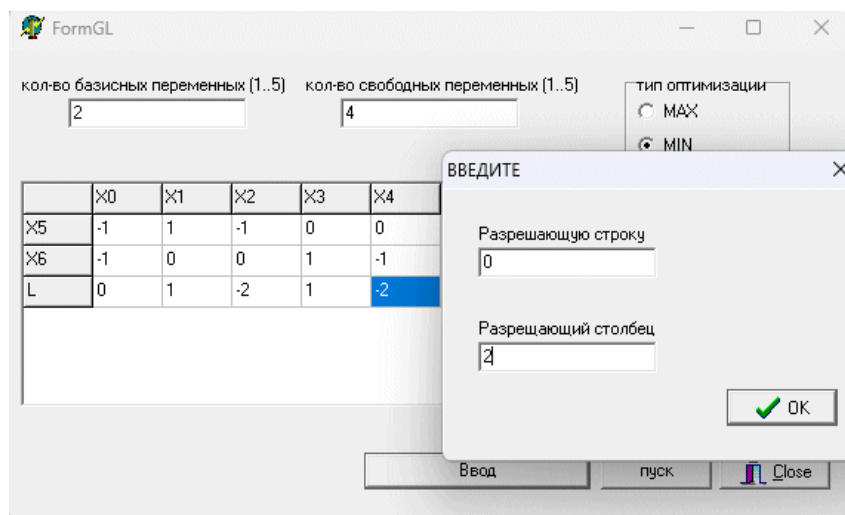


Рис. Д.1.7. Знімок зображення екрану, який виникає на першому кроці при знаходженні опорного рішення

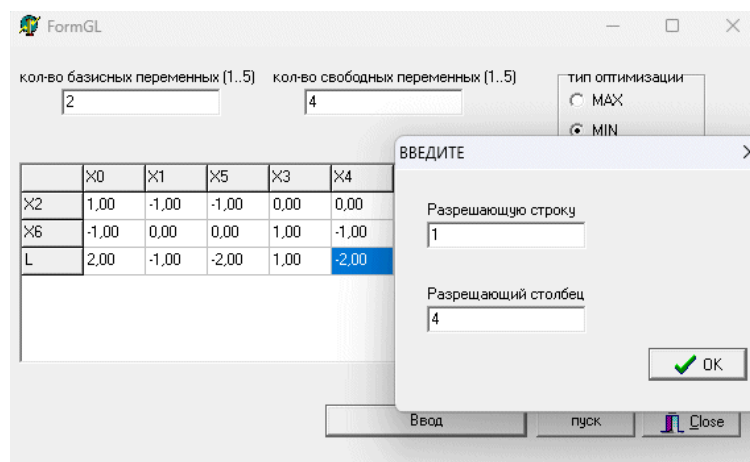


Рис. Д.1.8. Знімок зображення екрану, який виникає на другому кроці при знаходженні опорного рішення

FormGL

кол-во базисных переменных (1..5) кол-во свободных переменных (1..5)

тип оптимизации
☐ MAX
☒ MIN

	x0	x1	x5	x3	x6
x2	1,00	-1,00	-1,00	0,00	0,00
x4	1,00	0,00	0,00	-1,00	-1,00
L	4,00	-1,00	-2,00	-1,00	-2,00

Опорное решение получено

Оптимальное решение получено

Ввод пуск Close

Рис. Д.1.9. Знімок зображення екрану, який відповідає оптимальному рішенню

Отримане опорне рішення і є оптимальним. Оптимальне значення цільової функції дорівнює 4 і збігається з результатами, які були знайдені графічним методом та симплекс-методом для прямої задачі.

Приклад розв'язання транспортного завдання

Таблиця Д.2.1. Вхідна транспортна таблиця завдання з правильним балансом

$b_j \backslash a_i$	60	70	110
150	8	3	2
90	4	2	7

Таблиця Д.2.2. Побудова опорного рішення методом “північно-західного кута” та знаходження потенціалів постачальників та споживачів

$b_j \backslash a_i$	8 60	3 70	2 110
0 150	60 8	70 3	20 2
5 90	4	2	90 7

Система рівнянь для визначення потенціалів пунктів має вигляд:

$$\alpha_1=0; \alpha_1+\beta_1=8; \beta_1=8; \alpha_1+\beta_2=3; \beta_2=3; \alpha_1+\beta_3=2; \beta_3=2; \alpha_2+\beta_3=7; \alpha_2=5.$$

Різниці між вартостями та псевдовартостями перевезень за вільними комірками:

$$d_{21}=c_{21}-(\alpha_2+\beta_1)=4-(5+8)=-9; d_{22}=c_{22}-(\alpha_2+\beta_2)=2-(5+3)=-6.$$

Отримане опорне рішення не є оптимальним.

Повна вартість перевезень за отриманим опорним планом:

$$W = \sum_{j=1}^n \sum_{i=1}^m c_{ij}x_{ij} = 8 \cdot 60 + 3 \cdot 70 + 2 \cdot 20 + 7 \cdot 90 = 1360.$$

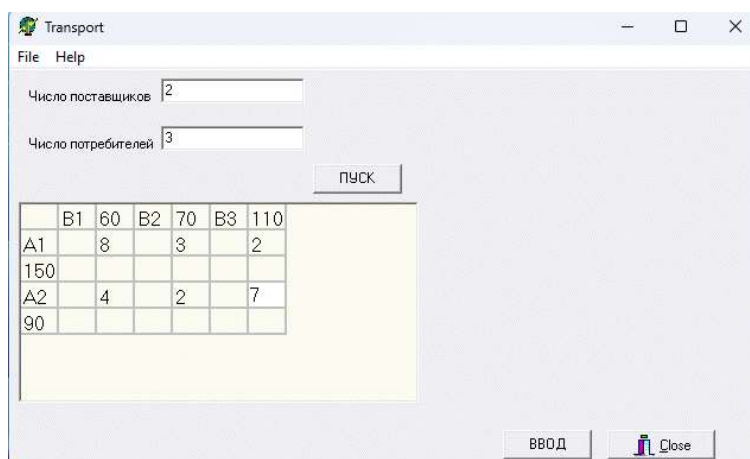


Рис. Д.2.1. Знімок зображення екрану, який відповідає вхідній транспортній таблиці

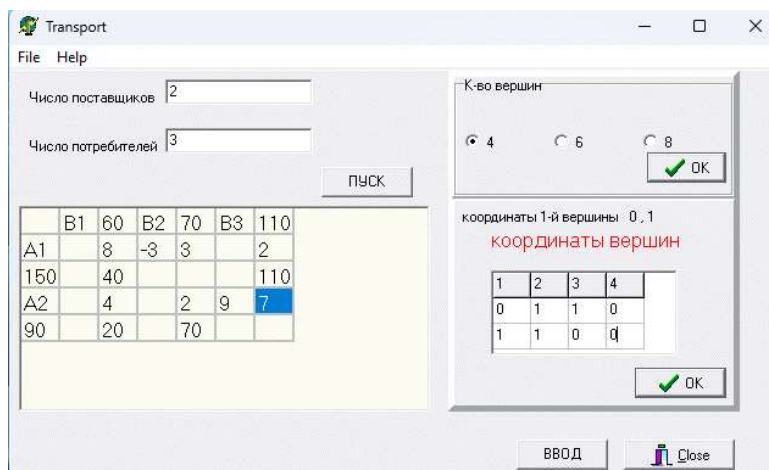


Рис. Д.2.2. Знімок зображення екрану, який відповідає опорному рішенню, що отриманий за методом “мінімального елемента”

Повна вартість перевезень за отриманим опорним планом:

$$W = \sum_{j=1}^n \sum_{i=1}^m c_{ij} x_{ij} = 8 \cdot 40 + 4 \cdot 20 + 2 \cdot 70 + 2 \cdot 110 = 760.$$

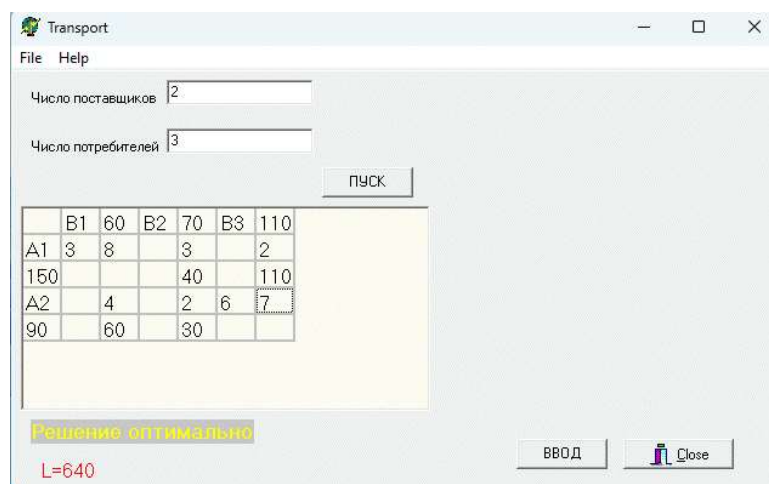


Рис. Д.2.3. Знімок зображення екрану, який відповідає оптимальному рішенню

Таблиця Д.2.3. Вхідна транспортна таблиця завдання з надлишком запасів

b_j	60	70	110	
a_i				
150	8	3	0	
90	4	2	0	

Останній стовбець в таблиці Д.2.3 відповідає фіктивному споживачу.

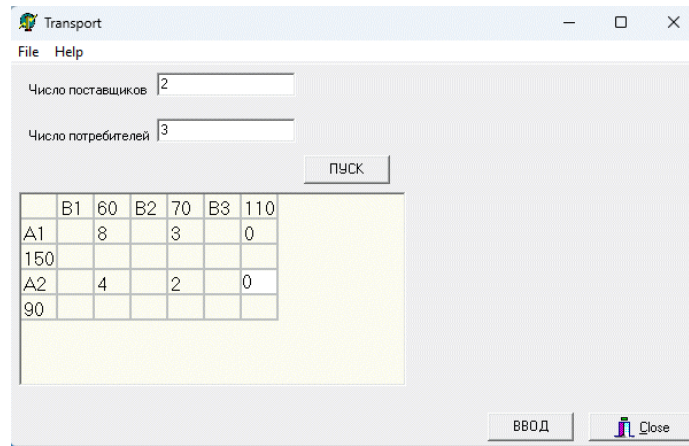


Рис. Д.2.4. Знімок зображення екрану, який відповідає вхідній транспортній таблиці

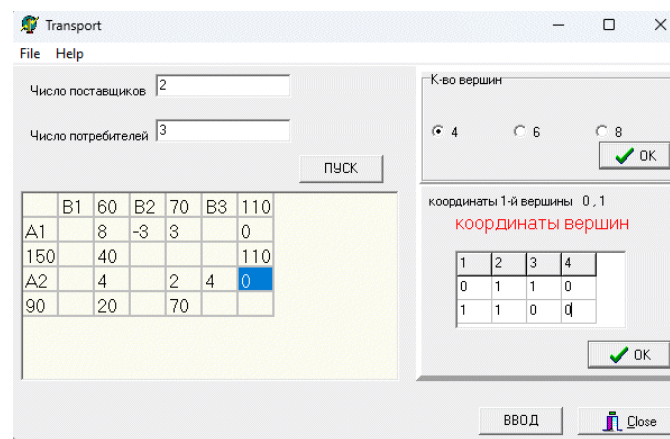


Рис. Д.2.5. Знімок зображення екрану, який відповідає опорному рішення, що отриманий за методом “мінімального елемента”

Повна вартість перевезень за отриманим опорним планом:

$$W = \sum_{j=1}^n \sum_{i=1}^m c_{ij} x_{ij} = 8 \cdot 40 + 4 \cdot 20 + 2 \cdot 70 = 540.$$

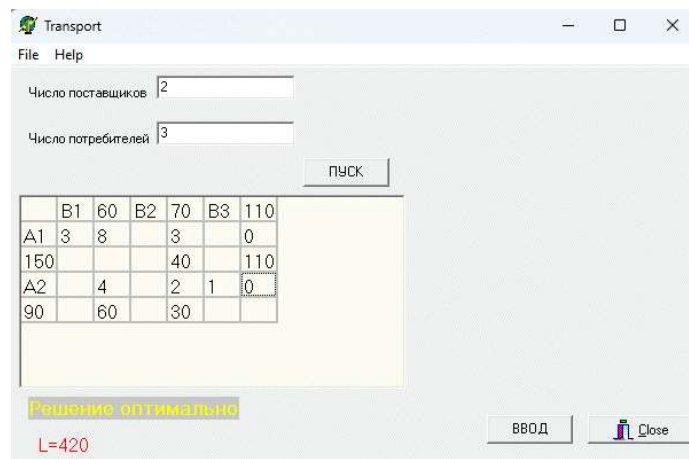


Рис. Д.2.6. Знімок зображення екрану, який відповідає оптимальному рішення

Висновки:

- опорний план перевезення, що отриманий за методом “мінімального елемента” краще ніж той, “північно-західного кута” про що свідчить менше значення повної вартості перевезень;
- жодний з отриманих опорних планів перевезення не є оптимальним і всі отримані оптимальні плани перевезення дають повні вартості перевезень нижчі ніж у відповідних їм опорних планів;
- опорний і оптимальний плани перевезення завдання з неправильним балансом дають меншу повну вартість перевезень ніж відповідні їм плани для завдання з правильним балансом, оскільки застосований метод зведення до правильного балансу передбачає розподіл надлишку запасів на користь фіктивного споживача, вартість транспортування до якого є нульовою.

Програма розв'язання завдання нелінійного програмування графічним методом

Для системи обмежень

$$\begin{cases} x^2 + y^2 \leq 9, \\ x \geq 0, y \geq 0, \\ y \geq 3, \end{cases} \quad (\text{Д.3.1})$$

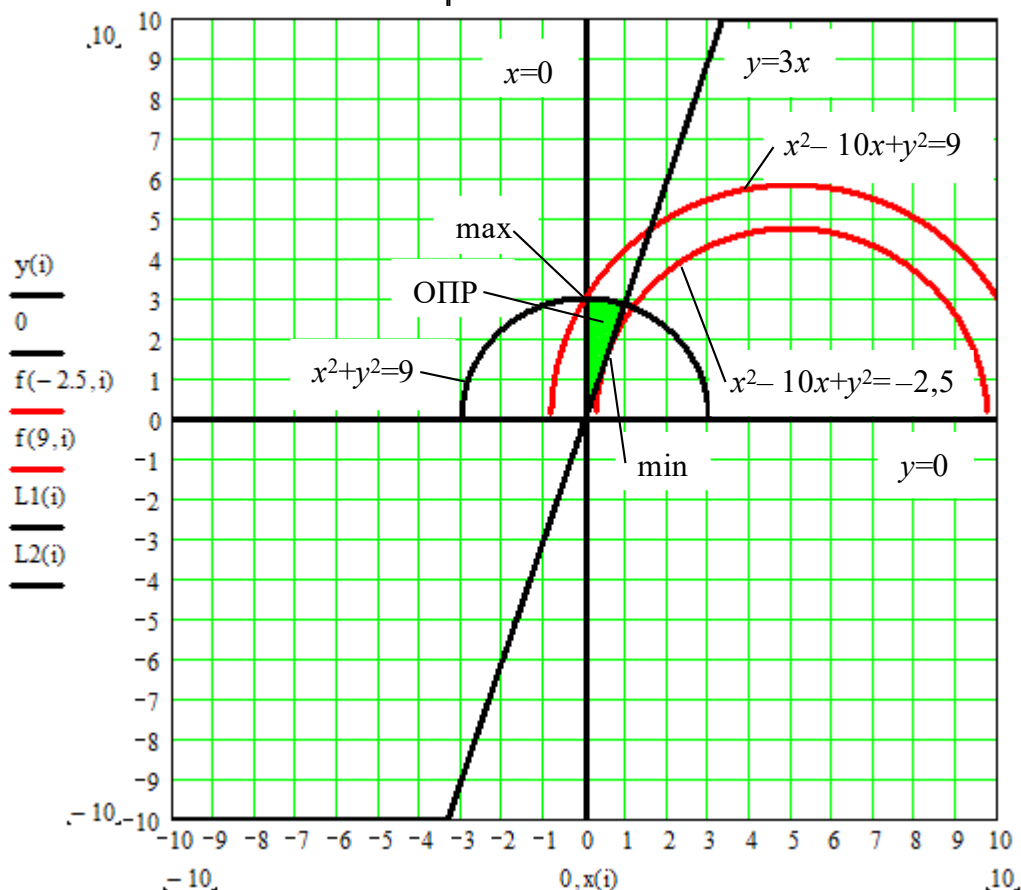
вирішуються завдання мінімізації та максимізації цільової функції

$$f(x, y) = x^2 - 10x + y^2 = (x - 5)^2 + y^2 - 25. \quad (\text{Д.3.2})$$

$$i := 0..2000 \quad x(i) := -10 + 0.01 \cdot i \quad y(j) := -10 + 0.01 \cdot j$$

$$f(h, i) := \sqrt{h + 25 - (x(i) - 5)^2}$$

$$L1(i) := \sqrt{9 - x(i)^2} \quad L2(i) := \begin{cases} 3 \cdot x(i) & \text{if } |3 \cdot x(i)| \leq 10 \\ i & \text{otherwise} \end{cases}$$



$$W(x,y) := x^2 - 10 \cdot x + y^2$$

$$x := 5$$

$$y := 5$$

Given

$$x^2 + y^2 \leq 9$$

$$y \geq 3 \cdot x$$

$$x \geq 0$$

$$y \geq 0$$

$$P := \text{Maximize}(W, x, y)$$

$$P = \begin{pmatrix} 0 \\ 3 \end{pmatrix}$$

Given

$$x^2 + y^2 \leq 9$$

$$y \geq 3 \cdot x$$

$$x \geq 0$$

$$y \geq 0$$

$$P := \text{Minimize}(W, x, y)$$

$$P = \begin{pmatrix} 0.5 \\ 1.5 \end{pmatrix}$$

Програма розв'язання завдання нелінійного програмування методом Лагранжа

Для обмежень

$$2x+5y-z=6, 7x+y+z=-6 \quad (\text{Д.4.1})$$

вирішуються завдання пошуку можливого мінімуму або максимуму цільової функції

$$W(x, y, z) = 6x^2 + x \cdot y - x \cdot z + 2y^2 - 4z^2. \quad (\text{Д.4.2})$$

$$x := 0 \quad y := 0 \quad z := 0 \quad \lambda_1 := 0 \quad \lambda_2 := 0$$

Given

$$12 \cdot x + y - 3 \cdot z + 2 \cdot \lambda_1 + 7 \cdot \lambda_2 = 0$$

$$x + 4 \cdot y + 5 \lambda_1 + \lambda_2 = 0$$

$$-3 \cdot x - 8 \cdot z - \lambda_1 + \lambda_2 = 0$$

$$2 \cdot x + 5y - z - 6 = 0$$

$$7 \cdot x + y + z + 6 = 0$$

$$\text{Find}(x, y, z, \lambda_1, \lambda_2) = \begin{pmatrix} -1.288 \\ 1.932 \\ 1.084 \\ -1.874 \\ 2.932 \end{pmatrix}$$

$$i := 0..40 \quad z(i) := -2 + 0.1 \cdot i \quad \Delta := -33$$

$$\Delta x(i) := z(i) + 6 + 5 \cdot (z(i) + 6) \quad \Delta y(i) := -2 \cdot (z(i) + 6) - 7 \cdot (z(i) + 6)$$

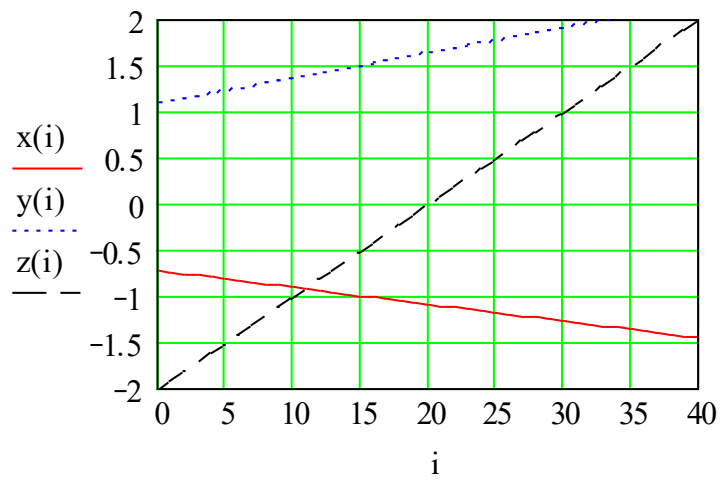
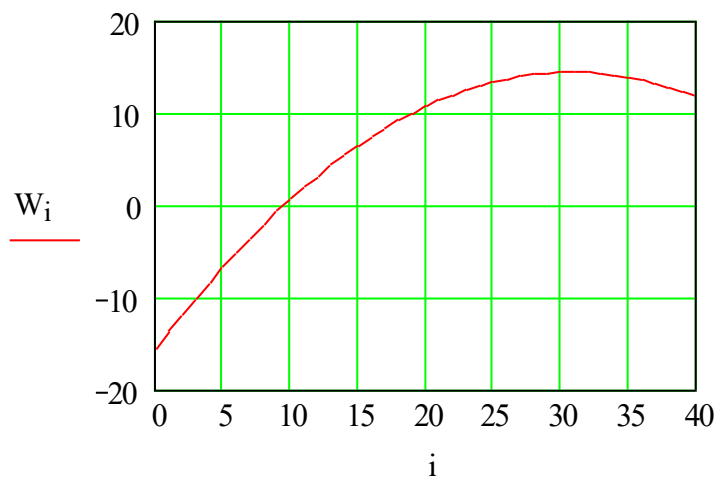
$$x(i) := \frac{\Delta x(i)}{\Delta} \quad y(i) := \frac{\Delta y(i)}{\Delta}$$

$$2 \cdot x(i) + 5y(i) - z(i) - 6 \quad 7 \cdot x(i) + y(i) + z(i) + 6$$

0
0
0
0
0
0
0
0
0
0

0
0
0
0
0
0
0
0
0
0

$$W_i := 6 \cdot x(i)^2 + x(i) \cdot y(i) - 3 \cdot x(i) \cdot z(i) + 2 \cdot y(i)^2 - 4 \cdot z(i)^2$$



Індивідуальні завдання до лабораторної роботи № 5

Платіжні матриці гри зі стихією

Варіанти № 1, 15

$A_i \setminus B_j$	B_1	B_2	B_3	B_4	B_5
A_1	5	7	4	2	3
A_2	1	5	4	1	4
A_3	4	2	2	2	3
A_4	1	6	6	5	8
A_5	4	5	8	2	3
A_6	7	1	3	3	8
A_7	4	5	7	2	3
p_j	0,25	0,1	0,15	0,4	0,1

Варіанти № 2, 16

$A_i \setminus B_j$	B_1	B_2	B_3	B_4	B_5
A_1	4	3	8	4	3
A_2	2	6	4	3	4
A_3	4	8	5	3	3
A_4	1	9	4	9	8
A_5	5	3	2	1	3
A_6	6	8	5	1	9
A_7	5	2	9	3	1
p_j	0,45	0,1	0,2	0,1	0,15

Варіанти № 3, 17

$A_i \setminus B_j$	B_1	B_2	B_3	B_4	B_5
A_1	1	4	9	2	2
A_2	5	1	5	6	5
A_3	3	6	3	4	2
A_4	1	7	3	6	3
A_5	2	5	1	5	3
A_6	3	9	5	4	4
A_7	6	2	4	6	7
p_j	0,2	0,2	0,2	0,3	0,1

Варіанти № 4, 18

$A_i \setminus B_j$	B_1	B_2	B_3	B_4	B_5
A_1	2	1	4	7	6
A_2	5	2	2	2	2
A_3	4	2	4	6	7
A_4	1	5	4	1	5
A_5	3	7	6	5	5
A_6	4	9	6	4	4
A_7	3	4	5	5	2
p_j	0,10	0,05	0,25	0,45	0,15

Варіанти № 5, 19

$A_i \setminus B_j$	B_1	B_2	B_3	B_4	B_5
A_1	9	6	2	5	8
A_2	5	4	1	2	5
A_3	2	1	1	1	2
A_4	4	1	5	6	4
A_5	3	5	4	5	1
A_6	5	6	4	4	2
A_7	1	3	1	3	5
p_j	0,25	0,4	0,05	0,1	0,2

Варіанти № 6, 20

$A_i \setminus B_j$	B_1	B_2	B_3	B_4	B_5
A_1	5	8	2	6	7
A_2	2	5	1	2	5
A_3	1	4	2	2	4
A_4	2	8	6	5	2
A_5	4	6	3	4	1
A_6	6	6	1	2	5
A_7	5	2	2	5	1
p_j	0,25	0,15	0,4	0,1	0,1

Варіанти № 7, 21

$A_i \setminus B_j$	B_1	B_2	B_3	B_4	B_5
A_1	4	5	5	2	7
A_2	1	2	2	6	4
A_3	1	3	9	4	2
A_4	5	6	2	5	5

A_5	6	4	2	9	2
A_6	7	6	3	2	1
A_7	1	5	4	5	1
p_j	0,2	0,45	0,05	0,2	0,1

Варіанти № 8, 22

$A_i \setminus B_j$	B_1	B_2	B_3	B_4	B_5
A_1	2	4	5	2	1
A_2	8	2	6	2	5
A_3	1	4	4	6	2
A_4	7	2	8	2	4
A_5	4	2	8	5	2
A_6	2	5	12	1	1
A_7	8	9	5	1	6
p_j	0,1	0,3	0,2	0,3	0,1

Варіанти № 9, 23

$A_i \setminus B_j$	B_1	B_2	B_3	B_4	B_5
A_1	4	4	2	1	3
A_2	7	4	1	2	1
A_3	5	2	3	4	5
A_4	4	7	6	1	2
A_5	2	3	5	7	5
A_6	2	1	6	1	2
A_7	4	4	2	2	1
p_j	0,45	0,15	0,2	0,05	0,15

Варіанти № 10, 24

$A_i \setminus B_j$	B_1	B_2	B_3	B_4	B_5
A_1	4	5	1	1	2
A_2	6	7	9	2	1
A_3	3	5	4	1	2
A_4	3	6	9	2	1
A_5	2	8	2	2	5
A_6	1	4	2	3	4
A_7	4	2	5	1	1
p_j	0,25	0,25	0,25	0,1	0,15

Варіанти № 11, 25

$A_i \setminus B_j$	B_1	B_2	B_3	B_4	B_5
A_1	4	5	5	2	7
A_2	1	2	2	6	4
A_3	1	3	1	4	2
A_4	5	6	2	5	5
A_5	4	5	8	2	3
A_6	7	1	3	3	8
A_7	4	5	7	2	3
p_j	0,2	0,3	0,2	0,1	0,2

Варіанти № 12, 26

$A_i \setminus B_j$	B_1	B_2	B_3	B_4	B_5
A_1	2	4	5	2	1
A_2	8	2	6	2	5
A_3	1	4	4	6	2
A_4	7	2	8	2	4
A_5	5	3	2	1	3
A_6	6	8	5	1	9
A_7	5	2	9	3	1
p_j	0,4	0,1	0,1	0,1	0,3

Варіанти № 13, 27

$A_i \setminus B_j$	B_1	B_2	B_3	B_4	B_5
A_1	4	4	2	1	3
A_2	7	4	1	2	1
A_3	5	2	3	4	5
A_4	4	7	6	1	2
A_5	2	5	1	5	3
A_6	3	9	5	4	4
A_7	6	2	4	6	7
p_j	0,3	0,15	0,3	0,15	0,1

Варіанти № 14, 28

$A_i \setminus B_j$	B_1	B_2	B_3	B_4	B_5
A_1	4	5	1	1	2
A_2	6	7	9	2	1
A_3	3	5	4	1	2
A_4	3	6	9	2	1

A_5	3	7	6	5	5
A_6	4	9	6	4	4
A_7	3	4	5	5	2
p_j	0,6	0,1	0,1	0,1	0,1

Платіжні матриці парної антагоністичної гри

Варіант № 1

$A_i \setminus B_j$	B_1	B_2	B_3
A_1	2	1	2
A_2	0	2	1

Варіант № 2

$A_i \setminus B_j$	B_1	B_2	B_3
A_1	2	4	3
A_2	3	5	2

Варіант № 3

$A_i \setminus B_j$	B_1	B_2	B_3
A_1	1	3	4
A_2	4	2	2

Варіант № 4

$A_i \setminus B_j$	B_1	B_2	B_3
A_1	4	2	6
A_2	2	3	8

Варіант № 5

$A_i \setminus B_j$	B_1	B_2	B_3
A_1	4	2	0
A_2	2	0	3

Варіант № 6

$A_i \setminus B_j$	B_1	B_2	B_3
A_1	1	2	3
A_2	3	3	2

Варіант № 7

$A_i \setminus B_j$	B_1	B_2	B_3
A_1	3	5	3
A_2	4	2	7

Варіант № 8

$A_i \setminus B_j$	B_1	B_2	B_3
A_1	1	2	3
A_2	2	0	1

Варіант № 9

$A_i \setminus B_j$	B_1	B_2	B_3
A_1	4	1	2
A_2	2	3	4

Варіант № 10

$A_i \setminus B_j$	B_1	B_2	B_3
A_1	2	5	4
A_2	4	2	1

Варіант № 11

$A_i \setminus B_j$	B_1	B_2	B_3
A_1	2	5	6
A_2	6	4	5

Варіант № 12

$A_i \setminus B_j$	B_1	B_2	B_3
A_1	7	8	9
A_2	8	9	4

Варіант № 13

$A_i \setminus B_j$	B_1	B_2	B_3
A_1	5	4	5
A_2	4	7	6

Варіант № 14

$A_i \setminus B_j$	B_1	B_2	B_3
A_1	0	4	4
A_2	5	0	1

Варіант № 15

$A_i \setminus B_j$	B_1	B_2
A_1	2	1
A_2	1	2
A_3	1	1

Варіант № 16

$A_i \setminus B_j$	B_1	B_2
A_1	1	4
A_2	3	2
A_3	1	3

Варіант № 17

$A_i \setminus B_j$	B_1	B_2
A_1	3	4
A_2	1	4
A_3	4	2

Варіант № 18

$A_i \setminus B_j$	B_1	B_2
A_1	4	1
A_2	2	2
A_3	2	3

Варіант № 19

$A_i \setminus B_j$	B_1	B_2
A_1	4	0
A_2	2	3
A_3	3	5

Варіант № 20

$A_i \setminus B_j$	B_1	B_2
A_1	0	4
A_2	3	2
A_3	2	2

Варіант № 21

$A_i \setminus B_j$	B_1	B_2
A_1	2	3
A_2	3	2
A_3	4	2

Варіант № 22

$A_i \setminus B_j$	B_1	B_2
A_1	5	6
A_2	4	4
A_3	2	7

Варіант № 23

$A_i \setminus B_j$	B_1	B_2
A_1	1	3
A_2	1	2
A_3	2	0

Варіант № 24

$A_i \setminus B_j$	B_1	B_2
A_1	4	1
A_2	2	3
A_3	3	4

Варіант № 25

$A_i \setminus B_j$	B_1	B_2
A_1	2	5
A_2	3	6
A_3	6	4

Варіант № 26

$A_i \setminus B_j$	B_1	B_2
A_1	7	9
A_2	9	5
A_3	8	4

Варіант № 27

$A_i \setminus B_j$	B_1	B_2
A_1	5	2
A_2	3	7
A_3	6	2

Варіант № 28

$A_i \setminus B_j$	B_1	B_2
A_1	1	2
A_2	5	0
A_3	0	1

Таблиця Д.6.1 – Вхідні дані для кластерного аналізу

Індекс варіанти	Діагноз	x1	x2	x3	x4	x5	x6	x7	x8
1	1	1	2	1	1	1	1	1	1
2	1	1	1	2	1	1	1	1	1
3	1	2	1	1	2	1	1	1	1
4	1	1	2	1	1	1	1	1	1
5	1	1	1	1	1	1	1	1	1
6	1	1	2	1	1	1	1	1	1
7	1	1	2	1	2	1	1	1	1
8	1	2	1	1	1	1	1	1	1
9	1	1	1	1	2	1	1	1	1
10	1	1	4	1	1	1	1	1	1
11	1	1	3	1	1	1	1	1	1
12	1	1	4	1	1	1	1	1	1
13	1	1	2	1	1	1	1	1	1
14	1	1	2	1	1	1	1	1	1
15	1	1	1	2	1	1	1	1	1
16	1	1	2	1	1	1	1	1	1
17	1	1	1	2	1	1	1	2	1
18	1	2	1	1	1	1	1	2	1
19	1	1	3	1	1	2	1	1	1
20	1	1	1	2	1	2	1	1	1
21	1	1	2	1	2	2	1	1	1
22	1	1	2	1	2	2	1	1	1
23	1	2	1	1	2	2	1	1	1
24	1	1	1	1	2	2	1	1	1
25	2	1	4	1	2	1	1	1	2
26	2	1	4	1	1	1	1	2	1
27	2	1	2	1	1	1	1	2	2
28	2	2	3	1	3	1	1	2	2
29	2	2	4	1	1	1	2	1	1
30	2	1	3	1	1	1	2	2	1
31	2	2	3	1	2	1	2	2	2
32	2	2	4	1	2	2	1	1	1
33	2	2	4	1	2	2	1	1	1
34	2	1	3	1	2	2	1	2	2
35	2	2	3	1	2	2	1	2	2
36	2	2	4	1	2	2	1	2	2
37	2	2	4	2	2	2	2	1	1
38	2	2	3	1	1	2	2	1	2
39	2	2	4	2	2	2	2	1	2
40	2	2	3	1	3	2	2	2	1
41	2	2	1	1	1	2	2	2	1
42	2	2	1	2	2	2	2	2	1
43	2	2	3	1	2	2	2	2	1
44	2	1	3	1	1	2	2	2	1
45	2	1	4	2	1	2	2	2	1
46	2	2	3	1	2	2	2	2	1
47	2	1	4	1	2	2	2	2	1
48	2	2	4	1	2	2	2	2	2
49	2	2	3	2	2	2	2	2	2
50	2	2	4	1	2	2	2	2	2
51	3	2	4	2	3	1	1	2	2
52	3	1	4	2	1	1	2	1	2

53	3	2	4	1	2	1	2	1	2
54	3	2	4	1	2	1	2	2	2
55	3	2	4	1	3	1	2	2	2
56	3	2	4	1	3	1	2	2	2
57	3	2	3	1	2	1	2	2	2
58	3	1	2	1	2	2	1	1	2
59	3	2	3	1	2	2	1	2	1
60	3	2	3	1	3	2	1	2	2
61	3	1	4	2	1	2	1	2	2
62	3	2	4	2	3	2	1	2	2
63	3	2	4	2	2	2	2	1	1
64	3	2	3	1	2	2	2	1	1
65	3	1	3	1	1	2	2	1	2
66	3	2	3	1	2	2	2	2	1
67	3	2	4	1	2	2	2	2	1
68	3	2	3	1	3	2	2	2	1
69	3	1	4	2	2	2	2	2	2
70	3	1	2	1	2	2	2	2	2
71	3	1	4	1	1	2	2	2	2
72	3	1	4	2	2	2	2	2	2
73	3	2	4	1	2	2	2	2	2
74	3	2	4	2	2	2	2	2	2
75	3	1	3	2	2	2	2	2	2
76	4	2	3	1	3	1	1	2	2
77	4	2	2	1	3	1	2	1	2
78	4	1	1	2	2	1	2	2	2
79	4	2	3	2	2	1	2	2	2
80	4	2	2	3	1	2	1	2	2
81	4	2	3	2	2	2	1	2	2
82	4	2	1	1	3	2	1	2	2
83	4	2	2	2	2	2	2	1	2
84	4	2	2	2	2	2	2	1	2
85	4	2	3	3	2	2	2	1	2
86	4	2	2	1	2	2	2	1	2
87	4	2	3	2	2	2	2	2	1
88	4	2	1	1	3	2	2	2	1
89	4	2	3	2	3	2	2	2	1
90	4	2	3	1	2	2	2	2	2
91	4	2	3	1	3	2	2	2	2
92	4	2	4	1	3	2	2	2	2
93	4	1	2	2	3	2	2	2	2
94	4	2	3	2	2	2	2	2	2
95	4	2	3	2	2	2	2	2	2
96	4	2	4	2	2	2	2	2	2
97	4	2	2	1	3	2	2	2	2
98	4	2	3	2	3	2	2	2	2
99	4	2	3	3	2	2	2	2	2
100	4	2	3	1	2	2	2	2	2
101	4	2	3	2	3	2	2	2	2
102	4	2	3	1	3	2	2	2	2
103	4	2	3	1	2	2	2	2	2

Програмні блоки кластеризації даних ітеративними методами групування

A - стандартизована матриця координат об'єктів в просторі ознак

$k_{\max}$ - число кластерів

n - число об'єктів

m - число ознак (симптомів за якими ставиться діагноз)

b - вектор розподілу об'єктів за кластерами на попередньому кроці ітерацій

b_1 - на вході в програмний блок - випадковий вектор початкового розподілу об'єктів за кластерами; в програмному блоці - вектор розподілу об'єктів за кластерами на поточному кроці ітерацій

M, M_1 - матриці приналежності об'єктів кластерам на попередньому та поточному кроках ітерацій

w - експонентна вага

e - похибка між нормою різниці матриць приналежностей на послідовних кроках при досягненні якої ітерації припиняються

count - кількість ітерацій

centr - матриця центроїдів кластерів, в якій перший стовбець - це кількість об'єктів в кластері (для методу K-means), а решта стовбців - координати центроїда в просторі ознак

S - норма різниці матриць приналежностей на останньому та передостанньому кроках ітерацій

Програмний блок кластеризації даних методом K-means

```

KMEANS(A, kmax, n, m, b1) :=
    count ← 0
    S ← 1
    "The condition for the end of iterations is the invariability"
    "of the distribution of objects by clusters at successive steps"
    while S > 0
        count ← count + 1
        S ← 0
        b ← b1
        for k ∈ 1..kmax
            for j ∈ 1..m
                centrk,j ← 0
            "Clusters centroids coordinates determining"
            for i ∈ 1..n
                for k ∈ 1..kmax
                    if bi = k
                        | centrk,1 ← centrk,1 + 1
                        | for j ∈ 2..m
                        | | centrk,j ← centrk,j + Ai,j-1
            for k ∈ 1..kmax
                for j ∈ 2..m
                    | centrk,j
                    centrk,j ←  $\frac{\text{centr}_{k,j}}{\text{centr}_{k,1}}$ 
            "Determining the new belonging of objects to clusters"
            for i ∈ 1..n
                | b1i ← 1
                | rmin ←  $\sqrt{\sum_{j=3}^m (\text{centr}_{1,j} - A_{i,j-1})^2}$ 
                | for k ∈ 2..kmax
                | | if rmin >  $\sqrt{\sum_{j=3}^m (\text{centr}_{k,j} - A_{i,j-1})^2}$ 
                | | | rmin ←  $\sqrt{\sum_{j=3}^m (\text{centr}_{k,j} - A_{i,j-1})^2}$ 
                | | | b1i ← k
            "Determination of the distance between the vectors of objects"
            "belonging to clusters at successive steps of iterations"
            S ←  $\sqrt{\sum_{i=1}^n (b1_i - b_i)^2}$ 
        (augmen(b, b1)
         centr
         count
    )

```

Програмний блок кластеризації даних методом Fuzzy C-means

```

CMEANS(A, kmax, n, m, M1, w, ε) :=
    count ← 0
    S ← 1
    while S > ε
        count ← count + 1
        S ← 0
        M ← M1
        for k ∈ 1..kmax
            for j ∈ 1..m
                centrk,j ← 0
        for k ∈ 1..kmax
            for j ∈ 1..m

$$\text{centr}_{k,j} \leftarrow \frac{\sum_{i=1}^n (M_{i,k})^w \cdot A_{i,j}}{\sum_{i=1}^n (M_{i,k})^w}$$

        for i ∈ 1..n
            for k ∈ 1..kmax

$$r_{k,i} \leftarrow \sqrt{\sum_{j=2}^m (\text{centr}_{k,j} - A_{i,j})^2}$$

            for k ∈ 1..kmax

$$M1_{i,k} \leftarrow \frac{1}{(r_{k,i})^{\frac{2}{w-1}} \cdot \sum_{p=1}^{kmax} \frac{1}{(r_{p,i})^{\frac{2}{w-1}}}} \quad \text{if } r_{k,i} \neq 0$$


$$M1_{i,k} \leftarrow 1 \quad \text{otherwise}$$


$$S \leftarrow \sqrt{\sum_{k=1}^{kmax} \sum_{i=1}^n (M1_{i,k} - M_{i,k})^2}$$

    (
        M1
        centr
        count
        S
    )

```

Програма кластеризації даних ієрархічними агломеративними методами з побудуванням дендрограми

Кластеризація здійснюється за всіма факторними ознаками всіх 103-х варіант таблиці

```
# Імпорт модуля pyplot бібліотеки matplotlib та бібліотек numpy, pandas
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
# Імпорт даних з файлу Excel
xls = pd.ExcelFile('c:\work\standard.xls')
df = pd.read_excel(xls, 'Standard', header=0)
data=np.array(df)
# Для кластеризації обираються дані всіх 103 рядки таблиці
X=data[0:102]
# Імпорт з бібліотеки SciPy функції побудування дендрограми dendrogram и функцій
# кластеризації різними методами: Уорда, повного та середнього зв'язку -
# ward, complete, average
from scipy.cluster.hierarchy import dendrogram, ward, complete, average
# Функція complete повертає масив зв'язків з відстанями,
# обчисленими з використанням евклідової метрикою в ході виконання
# агломеративної кластеризації методом повного зв'язку
linkage_array = complete(X)
# Побудування дендрограми
dendrogram(linkage_array)
# Позначки на дереві для відстаней, що відповідають чотирьом кластерам
ax = plt.gca()
bounds = ax.get_xbound()
ax.plot(bounds, [5.65, 5.65], '--', c='k')
ax.text(bounds[1], 5.65, ' Чотири кластери', va='center', fontdict={'size': 15})
# Підписи на осях
plt.xlabel("Індекс спостереження")
plt.ylabel("Міжкластерна відстань")
```

Програма реалізації класифікатора даних на основі багат шарового персептрону

```
# Імпорт бібліотеки mglearn надає можливість додаткової графічної інтерпретації
# результатів аналізу даних. Бібліотека mglearn не входить до
# середовища Anaconda за замовченням, тому її заздалегідь слід встановити командою
# з командного рядка pip install mglearn, або додати в програму рядок
#!pip install mglearn
# Імпорт бібліотеки mglearn та модуля pyplot бібліотеки matplotlib
import mglearn
import matplotlib.pyplot as plt
# Імпорт класів і функцій з модулів бібліотеки Scikit-Learn
from sklearn.model_selection import train_test_split
from sklearn.datasets import make_blobs
from sklearn.neural_network import MLPClassifier
from sklearn.datasets import make_moons
# Функція make_moons генерує набори точок такі, що в проекції на площину Оху
# виходять два півмісяці, вкладені один в одний. Параметр noise є
# середньоквадратичним відхиленням щільності ймовірності гаусового шуму. Чим він
# менше, тим менше точки відхиляються від значень, які визначаються функціями
# напівкіл. Параметр random_state повинен бути числом – початковим значенням
# генератора псевдовипадкових чисел. При використанні функції без вказівки
# параметру random_state кожного разу при запуску програми будується новий
# випадковий розподіл вхідного набору даних. Якщо ж параметр встановлено, при
# чому не важливо яке конкретно значення, розподіл буде збігатися з попереднім. Це
# корисно, якщо потрібні відтворювані результати, наприклад, при тестуванні, або
# документуванні. Вихідними параметрами є X – матриця ознак варіант – точок (їх
# горизонтальні та вертикальні координати) та y – вектор належності варіант до
# певного класу. Функція make_moons викликає функцію make_blobs, яка генерує
# набір даних. BLOB (Binary Large Object). Її парметрами є кількість ознак, якими
# характеризуються дані та кількість кластерів, до яких вони належать
X, y = make_moons(n_samples=100, noise=0.25, random_state=3)
# Функція train_test_split поділяє вхідний набір даних на навчальну і контрольну
# вибірки. Для якісного навчання імовірнісні розподіли навчальної і контрольної
# вибірок повинні бути близькі. Функція за замовчуванням автоматично перемішує
# дані, що забезпечує близькість вказаних розподілів. Параметр stratify встановлює
# ознаку даних за якою здійснюється контроль відповідності розподілів навчальної і
# контрольної вибірки за класами. У приведеному нижче випадку здійснюється
# стратифікація відповідно розподілу за ознакою y. Можна визначити або train_size
# або test_size – відсотки поділу об'ємів навчальної та контрольної (тестової) вибірок,
# але обидва вони не потрібні. Якщо явно встановлюються обидва параметри, вони
# повинні складати в сумі 1.
X_train, X_test, y_train, y_test = train_test_split(X, y, stratify=y, random_state=42)
```

```

#Імпорт бібліотеки numpy
import numpy as np
#Виведення розподілів навчальної і контрольної вибірки за класами.
print(f'Кількість рядків в y_train за класами: {np.bincount(y_train)}')
print(f'Кількість рядків в y_test за класами: {np.bincount(y_test)}')
#Клас MLPClassifier з використанням методу fit дозволяє здійснювати побудову і
#навчання нейронної мережі. За замовченням використовується функція активації
#ReLU. Створення змінної mlp як екземпляра класа MLPClassifier.
mlp = MLPClassifier(solver='lbfgs', max_iter=100000, random_state=0).fit(X_train, y_train)
#Алгоритм, який використовується для обчислення ваг синапсів задається за
#допомогою параметра solver. Алгоритм 'adam', використовуваний за замовчанням,
#вельми чутливий до масштабування даних (тому важливо перед класифікацією
#стандартизувати дані так, щоб кожна ознака мала середнє 0 і дисперсію 1). Інший
#алгоритм 'lbfgs' позбавлений цього недоліку, але є більш повільним і може зайняти
#багато часу в разі складних моделей або великих масивів даних. Параметр max_iter
#встановлює кількість циклів навчання.
#Візуалізація межі розділу класів, що побудована нейронною мережею на основі
#навчальної вибірки.
mglearn.plots.plot_2d_separator(mlp, X_train, fill=True, alpha=.3)
#Побудування крапок даних навчальної (або контрольної) вибірок
mglearn.discrete_scatter(X_train[:, 0], X_train[:, 1], y_train)
#mglearn.discrete_scatter(X_test[:, 0], X_test[:, 1], y_test)
#Виведення легенд осей
plt.xlabel("Ознака 0")
plt.ylabel("Ознака 1")

```

ЗМІСТ

1. ЗАГАЛЬНІ ПОЛОЖЕННЯ	3
2. ПОСТАНОВКА ЗАДАЧ ЛАБОРАТОРНИХ РОБІТ, МЕТОДИЧНІ РЕКОМЕНДАЦІЇ ДЛЯ ЇХ ВИКОНАННЯ ТА ІНДИВІДУАЛЬНІ ЗАВДАННЯ	4
2.1. Лабораторна робота № 1	
Прийняття рішень за допомогою методів лінійного програмування	4
2.2. Лабораторна робота № 2	
Прийняття рішень при розв'язанні транспортних завдань	13
2.3. Лабораторна робота № 3	
Розв'язання завдань нелінійного програмування графічним методом	23
2.4. Лабораторна робота № 4	
Розв'язання завдань нелінійного програмування методом Лагранжа	28
2.5. Лабораторна робота № 5	
Прийняття рішень в умовах невизначеності	32
2.6. Лабораторна робота № 6	
Прийняття рішень методами кластерного аналізу	46
2.7. Лабораторна робота № 7	
Ознайомлення з методами прийняття рішень за допомогою нейронних мереж	60
3. КРИТЕРІЇ ОЦІНЮВАННЯ	73
СПИСОК ВИКОРИСТАНОЇ ТА РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ	75
Додаток 1 Приклад розв'язання завдання лінійного програмування	76
Додаток 2 Приклад розв'язання транспортного завдання	82
Додаток 3 Програма розв'язання завдання нелінійного програмування графічним методом	86
Додаток 4 Програма розв'язання завдання нелінійного програмування методом Лагранжа	88
Додаток 5 Індивідуальне завдання до роботи № 5	90
Додаток 6 Вхідні дані до кластерного аналізу	99
Додаток 7 Програмні блоки кластеризації даних ітеративними методами групування	101
Додаток 8 Програма кластеризації даних ієрархічними агломеративними методами з побудуванням дендрограми	104
Додаток 9 Програма реалізації класифікатора даних на основі багат шарового персептрону	105

Навчальне видання

Заболотний Костянтин Сергійович
Кожевников Антон Вячеславович
Мещеряков Леонід Іванович
Соколова Наталя Олегівна

ТЕОРІЯ ТА МЕТОДИ ПРИЙНЯТТЯ РІШЕНЬ

Методичні рекомендації до виконання лабораторних робіт
для здобувачів ступеня магістра
спеціальностей галузі знань F Інформаційні технології

Видано в авторській редакції

Електронний ресурс.
Підписано до видання 04.11.2025. Авт. арк. 7,8.

Національний технічний університет «Дніпровська політехніка».
49005, м. Дніпро, просп. Дмитра Яворницького, 19.