

**Івано-Франківський національний технічний
університет нафти і газу**

Л.Р.Смоловик, , О.М.Витвицька

ДОСЛІДЖЕННЯ ОПЕРАЦІЙ

Конспект лекцій

2025

Л. Р. Смолоник, В.О.Витвицька

ДОСЛІДЖЕННЯ ОПЕРАЦІЙ

КОНСПЕКТ ЛЕКЦІЙ

**Івано-Франківськ
2025**

УДК 519.8(075.8)
С 51

Рецензент: Бандура А.І. доктор фізико-математичних наук,
зав. кафедри фізико-математичних наук, професор Івано-
Франківського національного технічного університету нафти і
газу

Рекомендовано методичною радою Івано-Франківського
національного технічного університету нафти і газу
(протокол № 13/1 від 26.05.2025),

Смоловик Л.Р., Витвицька О.М.

С 51 Дослідження операцій: Конспект лекцій.-Івано-
Франківськ: ІФНТУНГ, 2025, 139 с.

МВ-02070855-20802-2025

У конспекті лекцій викладено основи лінійної алгебри, лінійного та нелінійного програмування, елементи динамічного та сіткового програмування, елементи теорії ігор. Значну увагу приділяється економіко-математичним моделям та задачам економічного характеру. Пропонований конспект лекцій замінено та доповнено основними поняттями моделі та моделювання, прикладами розв'язування типових задач та питаннями для самоконтролю.

Конспект лекцій призначений для студентів економічних спеціальностей вищих навчальних закладів усіх форм навчання. Буде корисним для всіх спеціалістів, що

застосовують математичні методи в економіці.

УДК 519.8(075.8)

МВ 02070855-20802- 2025

Поняття моделі та моделювання

Модель – це основна категорія наукового пізнання людиною природи (від лат. Modulus — міра, зразок, аналог).
Модель – це будь-який образ певного оригіналу.

Моделювання – це вивчення яких-небудь явищ і процесів, або окремих об’єктів і їх систем шляхом побудови і вивчення їх моделей.

Головна мета моделювання – прогнозування поведінки моделі, залежно від зовнішніх факторів і прийняття

на цій основі рішення про методи управління оригіналом або його вдосконалення.

За фізичною природою моделі поділяються на:

фізична модель за своєю природою однакова з оригіналом (реальний клас);
аналогові моделі відмінні по своїй природі від оригіналу але однакові за формою законів, по яких функціонує модель і оригінал;
математичні моделі описують явища навколишнього світу за допомогою математичної символіки.

Математична модель – це наближене описання якого-небудь класу явищ зовнішнього світу за допомогою математичної символіки.

Відповідно **математичне моделювання** - це побудова, вивчення і використання математичних моделей.

Процес математичного моделювання включає в себе 4 основні етапи:

1. Формування законів, що описують основні об'єкти моделей

Цей етап потребує глибокого знання всіх факторів, що відносяться до даного явища і широкого розуміння їх взаємозв'язків. Він закінчується записом сформульованих якісних уявлень про зв'язки між окремими об'єктами моделі у математичних термінах.

2. Дослідження математичної моделі

Основне завдання – розв'язок прямої задачі, тобто одержання в результаті теоретичного аналізу моделі вихідної інформації для співвідношення її з результатами спостережень за явищем, що вивчається, тобто за оригіналом.

На цьому етапі важливу роль відіграє математичний апарат, що потрібен для аналізу моделі, а також

обчислювальна техніка.

3. Перевірка адекватності моделі

Це з'ясування на скільки задовольняє прийнята модель критерії практики; чи узгоджуються теоретичні наслідки моделі з оригіналом в межах точності спостережень. Якщо відходження виходить за межі точності, то така модель повинна бути відкинута.

4. Аналіз і уточнення моделі

Проводиться в зв'язку з покращенням даних про досліджуване явище з метою поліпшення його прогнозних можливостей.

Предмет дослідження операцій

Дослідження операцій (ДО) — це дисципліна, що займається розробкою й застосуванням методів знаходження оптимальних рішень на основі математичного моделювання у різних областях людської діяльності. ДО тісно пов'язане з системним аналізом, математичним програмуванням, теорією оптимальних рішень. ДО — це застосування математичних, кількісних методів для обґрунтування рішень у всіх галузях цілеспрямованої людської діяльності. ДО починається тоді, коли для обґрунтування рішень використовується той чи інший математичний апарат

2 Метод Жордана-Гаусса розв'язування систем лінійних алгебраїчних рівнянь

Розглянемо систему лінійних рівнянь з невідомими:

Випишемо розширену матрицю системи виду:

.

Суть методу Жордана-Гаусса полягає в тому, щоб звести систему до базисної форми виду

за допомогою елементарних перетворень рівнянь системи.

При цьому змінні є базисними, а — вільними. В системі базисним змінним відповідають одиничні базисні вектори:

; ; ... ; .

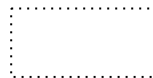
З системи визначимо базисні змінні:

Система (3) задає загальний розв'язок системи (1), де вільні змінні можуть бути довільними числами.

Частинний розв'язок, одержаний з (3) при називається **базисним** розв'язком. Система може мати декілька базисних розв'язків. Базисний розв'язок, у якого всі змінні невід'ємні, називається **допустимим базисним** розв'язком.

Алгоритм методу Жордана-Гаусса.

1. Вибрати провідний елемент .
2. Поділити всі елементи провідного рядка на , а всі елементи провідного стовпця замінити нулями.
3. Всі інші елементи розширеної матриці системи знайти за правилом прямокутника: щоб знайти елемент , треба від елемента відняти добуток коефіцієнтів, які стоять навпроти нього у провідних стовпці і рядку, поділений на провідний елемент, розміщений по діагоналі від елемента :



4. Щоб виділити наступну базисну змінну, обрати провідним ненульовий елемент у будь-якому рядку (крім -го) і повторити кроки 1-3.
Базисних змінних вибирають стільки, який ранг системи

векторів

Зауваження 1. Якщо в процесі виключень з'явиться рівняння, у якому ліва частина дорівнює нулю, а вільний член відмінний від нуля – це ознака несумісності системи.

Зауваження 2. Якщо ліва і права частина деякого рівняння перетворюється в нуль, то воно є лінійною комбінацією решти, і його треба виключити з розгляду. Таким чином, в процесі жорданових перетворень (застосовуються як у випадку , так і у випадку) або встановлюється несумісність системи, або знаходяться її розв'язки.

Приклад 1.1

Розв'язати систему:

Скористаємося методом Жордана-Гаусса, записуючи результати перетворень у вигляді таблиць





-
- базисні змінні;
 - вільна змінна.

Відповідь: .

2 Математичні моделі економічних задач

Задачі математичного програмування діляться на:

лінійні (в яких - лінійні функції) та *нелінійні* (у іншому випадку);
стохастичні (враховується випадковий характер економічних процесів) і *детерміновані* (враховують лише усереднені значення параметрів, не містять випадкових змінних і параметрів);
динамічні (розглядається розвиток економічної системи)

протягом декількох періодів часу) і *статичні* (розглядається один період часу).

Розвиток математичного програмування розпочався у 30-х роках з виходом у світ праць радянського вченого Л.В.Канторовича. Лінійне програмування оформилося як окремий розділ прикладної математики в 40-50-х роках ХХ ст. і вважається повністю дослідженим. Тепер 80% всіх задач оптимізації, що розв'язуються на практиці, є задачами лінійного програмування. За досягнення у цій галузі були присуджені нобелівські премії.

2.1 Лінійне програмування

Лінійне програмування – це наука про методи відшукування найбільших та найменших значень лінійної цільової функції при обмеженнях, що задаються системою лінійних нерівностей, рівнянь, або рівнянь і нерівностей.

Наприклад, задача лінійного програмування може мати вигляд:

;

Де – цільова функція, – параметри задачі,
– керовані змінні, (2) – система обмежень.

Кожен розв'язок системи (2) називається **допустимим розв'язком** (або **допустимим планом**). Допустимий розв'язок, який дає максимум цільової функції, називається **оптимальним розв'язком** задачі (1)-(2).

Зауважимо, що у системі (2) можуть бути нерівності замість деяких або всіх рівностей. Крім цього, можуть задаватися умови на невід'ємність керованих змінних.

Звичайно, математична модель не може врахувати всі особливості операцій. Вона вдала, якщо з одного боку, адекватно відображає суть задачі, з іншого – досить проста для розв’язання.

Математичні моделі дослідження операцій можна класифікувати в залежності від призначення та змісту модельованої операції, сенсу розв’язуваної проблеми, використовуваного математичного апарату.

2.2 Постановка задач лінійного програмування.

Побудова математичних моделей деяких економічних задач

Приклад 2.1 Задача про використання сировини

Для виготовлення двох видів продукції і використовуються три види сировини , , .

Дані виробництва наведені у таблиці:

Вид сировини	Кількість одиниць сировини на виготовлення одиниці продукції		Запаси сировини
	2	5	20
	8	5	40
	5	6	30
Прибуток від одиниці продукції	50	40	

Скласти такий план випуску продукції, щоб при її реалізації одержати максимальний прибуток.

Розв’язання.

Нехай - кількість одиниць продукції , – продукції

Тоді, враховуючи кількість одиниць сировини, що витрачається на виготовлення одиниці продукції, а також

запаси сировини, одержимо систему обмежень

Тобто, кількість сировини, що витрачається на виробництво продукції, не повинна перевищувати запасів.

Якщо продукція не випускається, то , інакше .
Отже, обмеження на змінні додаються такі:

Реалізація одиниць продукції дає прибуток 50 ,
реалізація одиниць продукції – 40 , тому сумарний
прибуток .

Отже, математична модель задачі про використання сировини така:

;

Зауваження. Задачу про використання сировини можна поширити на випадок видів продукції і видів сировини:

;

Приклад 2.2 Задача про розкрій.

З листового прокату треба вирізати заготовки двох типів і для виробництва 60 шт виробів. Для одного виробу треба 3 заготовки типу і 8 заготовок типу . Є чотири варіанти , , , розкрою листа, які задані таблицею.

Заготовка	Варіант розкрою				Потреби
	4 0	3 4	2 6	1 10	3*60=180 8*60=480
Відходи	12	5	3	0	

Скласти оптимальний план розкрою, тобто вказати, скільки листів треба розкрюювати згідно кожного варіанту, щоб одержати необхідну кількість заготовок кожного типу при мінімальних відходах.

Розв'язання.

Нехай - кількість листів для розкрою варіантом ,

Математична модель задачі

;

Складніші постановки можуть включати вимоги цілочисельності , відображати наявність листів різного

формату, враховувати інші корисні якості.

Узагальнена задача матиме вигляд

;

Приклад 2.3 Задача про складання кормового раціону

При відгодівлі кожна тварина повинна щоденно отримувати не менше 9 одиниць поживної речовини S_1 , не менше 8 одиниць речовини S_2 і не менше 12 одиниць речовини S_3 . Для складання раціону використовують два види корму P_1 і P_2 . Вміст поживних речовин в кормах і вартість кормів наведено у таблиці.

Скласти добовий раціон потрібної поживності, причому витрати повинні бути мінімальні

Поживні речовини	Кількість одиниць поживної речовини в 1 кг корму		Необхідна кількість поживних речовин
	3	1	9
	1	2	8
	1	6	12
Вартість 1 кг корму	4	6	

Розв'язання.

Нехай x_1 і x_2 – кількість корму P_1 і P_2 відповідно в добовому раціоні.

Математична модель матиме вигляд

;

Задачу також можна узагальнити на випадок кормів і поживних речовин:

;

, , , , , .

3 Різні форми запису задач лінійного програмування. Перехід від однієї форми запису до іншої

За формою запису серед задач лінійного програмування виділяють: *стандартні* (або *симетричні*), *канонічні*, *загальні*.

Стандартна задача – це задача дослідження цільової функції на максимум з обмеженнями – нерівностями типу „ ” і невід’ємними змінними. Вона має вигляд:

;

Задачу можна подати за допомогою:

а) матричного запису:
$$\begin{cases} \sum_{j=1}^n a_{ij}x_j = b_i, & i=1, \dots, m \\ x_j \geq 0, & j=1, \dots, n \end{cases}$$
 де $A = (a_{ij})$; $b = (b_i)$; $x = (x_j)$;

б) векторного запису:
$$\begin{cases} Ax = b \\ x \geq 0 \end{cases}$$
 де $A = (a_{ij})$; $b = (b_i)$; $x = (x_j)$;

Канонічна задача – це задача максимізації цільової функції при обмеженнях-рівностях, причому всі змінні невід’ємні. У канонічній задачі кількість невідомих завжди більша за кількість рівнянь.

Загальний вигляд канонічної задачі:

$$\begin{cases} \sum_{j=1}^n a_{ij}x_j = b_i, & i=1, \dots, m \\ x_j \geq 0, & j=1, \dots, n \end{cases}$$

Частинним випадком канонічної задачі є задача в базисній формі, тобто задача - , у якої і матриця системи обмежень містить одиничну матрицю розмірністю . Задача має вигляд:

$$\begin{aligned} & \text{;} \\ & \text{;} \\ & \text{;} \end{aligned}$$

Загальною задачею лінійного програмування є задача, у якій в систему обмежень входять як рівності, так і нерівності, вимога невід'ємності не накладається на усі змінні. Стандартна і канонічна задачі входять в клас загальних задач. Загальна задача має вигляд:

$$\begin{aligned} & \text{;} \\ & \text{,} \text{;} \\ & \text{,} \text{;} \\ & \text{,} \text{;} \\ & \text{,} \text{,} \end{aligned}$$

-довільні,

На практиці більшість економічних задач зводяться до побудови стандартних і загальних задач. Однак, основні методи розв'язування розроблені для канонічних задач. Тому виникає потреба перетворювати задачі лінійного програмування з однієї форми в іншу. Задачу мінімізації можна формально замінити на задачу максимізації і навпаки за допомогою рівностей:

;

Основним залишається перетворення системи обмежень.

Нерівності (9) множенням лівих і правих частин на (-1) можна перетворити в нерівності (8) і навпаки. Якщо в задачі деяка змінна довільного знаку, то її подають у вигляді різниці

двох невід'ємних змінних: , де і . У випадку, коли деяка змінна , то її замінюють на

, де

Існують такі способи перетворень.

1. Перехід від стандартної до канонічної форми здійснюється шляхом введення додаткових невід'ємних змінних так, щоб система нерівностей перетворилася в систему рівнянь:

При цьому нова система обмежень буде мати рівнянь і змінних.

Якщо є розв'язком канонічної задачі, то є розв'язком стандартної задачі.

2. Перехід від канонічної до стандартної форми можна здійснити двома способами. Першим – кожену рівність замінити однією з систем нерівностей:

Другий метод полягає у зменшенні кількості невідомих за допомогою методу Жордана-Гаусса: в системі обмежень канонічної задачі перетворимо змінні у базисні. Одержимо систему, з якої знайдемо базисні змінні:

(13)

оскільки , , то покладемо , тоді одержимо:

Підставимо систему $\begin{cases} x_1 + x_2 = 10 \\ x_1 - x_2 = 2 \end{cases}$ в цільову функцію, одержимо

Таким чином, одержали задачу в стандартному вигляді.

Приклад 3.1

Звести до канонічного виду задачу лінійного програмування.

;

;

Відомо, що $x_1, x_2 \geq 0$;

змінні: x_1, x_2 і x_3 ;

;

Крім того, введемо дві додаткові змінні x_4 і x_5 , щоб нерівності перетворити в рівності.

Задача набуде вигляду:

;

$x_1, x_2, x_3, x_4, x_5 \geq 0$

x_1, x_2, x_3, x_4, x_5

Приклад 3.2

Звести до стандартного виду задачу лінійного програмування.

;

Зменшимо кількість невідомих. Для цього виділимо базис методом Жордана-Гаусса:

$$\begin{array}{rrrrr}
 1 & 2 & 1 & 1 & 5 \\
 2 & 3 & 1 & 3 & 9 \\
 \hline
 1 & 2 & 1 & 1 & 5 \\
 0 & -1 & -1 & 1 & -1 \\
 \hline
 1 & 2 & 1 & 1 & 5 \\
 0 & 1 & 1 & -1 & 1 \\
 \hline
 1 & 0 & -1 & 3 & 3 \\
 0 & 1 & 1 & -1 & 1
 \end{array}$$

Відповідь: ;

;

4 Властивості задач лінійного програмування

4.1 Опуклі множини

Введемо деякі поняття з теорії множин.

Розглянемо точки A, B . Точка C лежить на відрізку AB , якщо вектори $\vec{AC}$ і $\vec{AB}$ колінеарні, тобто їх координати пропорційні:

;

або

де λ, μ — деякі числа.

Можна записати: $C = \lambda A + \mu B$, (1)

де $\lambda + \mu = 1$.

Означення 1. Точка C , для якої виконується рівність називається **опуклою лінійною комбінацією точок** A і B . Точки A і B називаються **кутовими** точками відрізка AB . Очевидно, що кутові точки не є лінійними комбінаціями жодних двох точок відрізка.

Зауважимо, що співвідношення $C = \lambda A + \mu B$ правильне незалежно від розмірності простору.

Означення 2. Точка C є **опуклою лінійною комбінацією точок** $A_1, A_2, \dots, A_n$, n -вимірного простору, якщо існують

числа $\lambda_1, \lambda_2, \dots, \lambda_n$, такі що $\sum_{i=1}^n \lambda_i = 1$ і

Означення 3. Множина точок простору називається **опуклою**, якщо вона разом з довільними своїми двома точками містить і їх довільну опуклу лінійну комбінацію.

Геометрично це означає, що опукла множина разом з довільними своїми двома точками повинна містити і відрізок, що їх з'єднує.

Опуклі множини: відрізок, пряма, півплощина, круг,

квадрат, куля, піраміда.

Прикладами неопуклих множин є парабола, коло, еліпс, а також:

Твердження 1. Перетин будь-якої кількості опуклих множин є опуклою множиною.

Означення 4. Точка опуклої множини називається **кутовою**, якщо вона не може бути зображена у вигляді опуклої лінійної комбінації будь-яких двох різних точок цієї множини.

Означення 5

1) Множина називається **відкритою**, якщо вона містить тільки свої внутрішні точки. Якщо множина, крім внутрішніх, містить всі граничні точки, то вона називається **замкнутою**.

2) Множина називається **обмеженою**, якщо існує куля скінченного розміру, яка повністю містить цю множину. Інакше множина необмежена.

Означення 6

1) **Опуклим многокутником** називається опукла замкнена обмежена множина на площині, яка має скінчене число кутових точок.

2) Кутові точки многокутника називаються його **вершинами**, а відрізки, що з'єднують дві вершини і утворюють його границю, - **сторонами многокутника**.

3) **Опорною прямою** опуклого многокутника називається пряма, яка має з многокутником, розміщеним по один бік від неї, принаймні одну спільну точку.

На рис.4.1 ABCDE – опуклий многокутник; AE, CF – опуклі прямі.

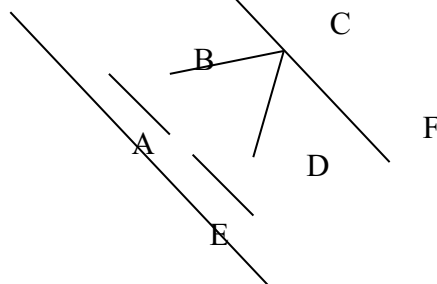


Рисунок 4.1

Означення 7

1) **Опуклим многогранником** називається замкнена обмежена опукла множина - вимірного простору, яка має скінчене число кутових точок.

2) Кутові точки многогранника називаються **вершинами**; многокутники, що обмежують многогранники – **гранями**; відрізки, по яких перетинаються грані – **ребрами**.

3) **Опорною площиною многогранника** називається площина, яка має з многогранником, розміщеним по один бік від площини, принаймні одну спільну точку.

Твердження 2 Опуклий многокутник (многогранник) є опуклою лінійною комбінацією своїх кутових точок

‘ ‘ ‘ ‘

4.2 Геометричне тлумачення задачі лінійного програмування

Розглянемо симетричну задачу лінійного програмування (тобто з обмеженнями – нерівностями):

;

;

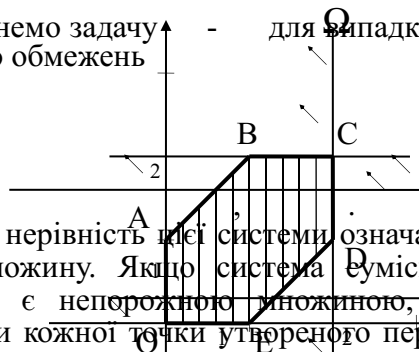
‘ ‘

Сукупність чисел , які задовольняють

систему обмежень , , називають розв'язком системи обмежень або **планом**.

Якщо існує хоча б один план, то система обмежень називається **сумісною**, інакше – вона несумісна.

Розглянемо задачу - для випадку , тобто задачу з системою обмежень



Кожна нерівність цієї системи означає півплощину, тобто опуклу множину. Якщо система сумісна, то перетин цих півплощин є непорожньою множиною, яка також опукла. Координати кожної точки утвореного перетину є розв'язками системи обмежень. Сукупність всіх цих точок називається многокутником розв'язків і позначається .

Наприклад, система обмежень має зображений многокутник розв'язків:

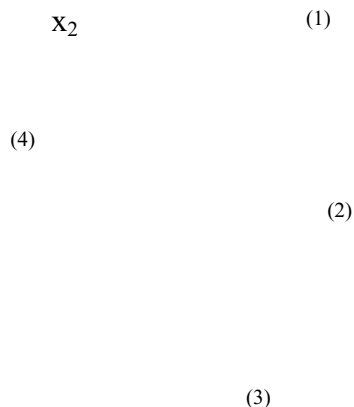


Рисунок 4.2

Многокутник розв'язків може бути точкою, відрізком, променем, многокутником, необмеженим многокутником.

У випадку сукупність розв'язків системи обмежень називається многогранником розв'язків.

Отже, геометрично симетрична (стандартна) задача лінійного програмування полягає у знаходженні такої точки многокутника (многогранника) розв'язків, координати якої надають цільовій функції максимального значення. При цьому допустимими розв'язками є всі точки многогранника розв'язків.

4.3 Властивості розв'язків задачі лінійного програмування

Розглянемо канонічну задачу у векторній формі

$$\begin{aligned} & \text{Знайти } \vec{x} \text{, що} \\ & \vec{A} \vec{x} = \vec{b}, \quad \vec{x} \geq \vec{0}, \end{aligned}$$

де $\vec{A} = (a_{ij})$; $\vec{b} = (b_i)$; $\vec{x} = (x_j)$, $i = \overline{1, m}$, $j = \overline{1, n}$.

Запис означає, що $\vec{x}$ є лінійною комбінацією векторів

.

Означення 1 План задачі, називається **опорним**, якщо система векторів-стовпчиків

які відповідають додатним координатам, лінійно незалежна.

Оскільки вектори мають по координат, то лінійно незалежними можуть бути не більше координат. Це означає, що в опорному плані є не більше m строго додатних координат.

Базисні розв'язки системи з невід'ємними координатами є опорними планами.

Приклад 4.1 Перевірити, чи план є опорним, якщо система обмежень така

$$\begin{cases} x_1 + 2x_2 + 3x_3 = 12 \\ x_1 + x_2 + x_3 = 6 \\ x_1, x_2, x_3 \geq 0 \end{cases}$$

Додатним координатам плану i відповідають вектори

Перевіримо, чи вони лінійно незалежні. Для цього знайдемо ранг матриці

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 1 & 1 \end{pmatrix}$$

обчислимо визначник.

Отже, вектори i лінійно незалежні, а план - опорний.

Означення 2. Опорний план називається

невиродженим, якщо число його додатних координат рівне m , і виродженим, якщо число додатних координат менше m .

У прикладі план невірджений.

Нехай ранг матриці дорівнює .
Означення 3. Базисом опорного плану називається

система з лінійно незалежних векторів-стовпчиків матриці , яка містить всі вектори, що відповідають додатним координатам опорного плану.

У прикладі вектори і утворюють базис опорного

плану

Таким чином, базис невірдженого опорного плану визначається однозначно, а у виродженого їх може бути декілька.

Запишемо задачу - у вигляді

;

;

;

і розглянемо основні властивості розв'язків цієї задачі.

Теорема 1. Множина всіх розв'язків системи опукла, якщо вона непорожня.

Доведемо, що якщо , – розв'язки задачі - , то їх лінійна комбінація опукла:

,

, , .

Оскільки , – плани, то , , .
 , . Тоді

Отже, $\bar{x}$ задовольняє систему $Ax \leq b$.

Оскільки, $\bar{x} \in X$, $\bar{x} \in \{x \in X \mid Ax \leq b\}$, то $\bar{x} \in X \cap \{x \in X \mid Ax \leq b\}$, тобто виконується $\bar{x} \in X \cap \{x \in X \mid Ax \leq b\}$.

Опуклу множину планів задачі лінійного програмування позначимо P . Можна довести, що P буде опуклим многокутником (обмеженим чи необмеженим).

Теорема 2. Якщо цільова функція $f(x)$ має максимум на опуклому многограннику допустимих розв'язків P , то він досягається у вершині цього многогранника.

Нехай многогранник P обмежений і має скінченне число кутових точок $v_1, v_2, \dots, v_k$. Точку, в якій $f(x)$ досягає максимуму, позначимо x^* .

Якщо серед вершин $v_1, v_2, \dots, v_k$, є хоч одна, в якій досягає $f(x)$, то теорема доведена.

Нехай x^* не є кутовою точкою. Тоді x^* є внутрішньою точкою P .

Якщо P – не кутова, то вона є опуклою лінійною комбінацією кутових, тобто $x^* = \sum_{i=1}^k \lambda_i v_i$, де $\lambda_i \geq 0$, $\sum_{i=1}^k \lambda_i = 1$.

Тоді $f(x^*) = \sum_{i=1}^k \lambda_i f(v_i)$.

Отже, $f(x^*) = \sum_{i=1}^k \lambda_i f(v_i)$ – суперечність. Тому x^* – одна з кутових точок P .

Отже, екстремум цільової функції досягається в вершині многогранника розв'язків, але не обов'язково у одній точці.

Теорема 3. Якщо цільова функція має максимум більше,

ніж в одній вершині, многогранника P , то цей максимум досягається у будь-якій точці, яка є опуклою лінійною комбінацією цих вершин.

Нехай цільова функція f досягає максимуму у вершинах $v_1, v_2, \dots, v_k$ многогранника P , тобто $f(v_i) = M$, $i = 1, 2, \dots, k$. Тоді для довільної прямої лінійної комбінації цих вершин

$$M = \alpha_1 f(v_1) + \alpha_2 f(v_2) + \dots + \alpha_k f(v_k) = f(\alpha_1 v_1 + \alpha_2 v_2 + \dots + \alpha_k v_k) = f(v),$$

де $\alpha_i \geq 0$, $\alpha_1 + \alpha_2 + \dots + \alpha_k = 1$, $v = \alpha_1 v_1 + \alpha_2 v_2 + \dots + \alpha_k v_k$ є вершиною многогранника P .

Отже, для знаходження максимуму цільової функції досить дослідити всі вершини многогранника P .

Теорема 4. Вектор x є опорним планом задачі (P, f) тоді і лише тоді, коли x є вершиною многогранника розв'язків P .

З теореми випливає, що алгебраїчне поняття опорного плану еквівалентне до геометричного поняття вершини многогранника розв'язків.

Наслідок 1. Кожна вершина многогранника P має не більше ніж n додатних координат.

За означенням опорного плану, x є опорним планом, якщо x має не більше ніж n додатних координат.

Наслідок 2. Кожній вершині многогранника P відповідає лінійно-незалежних розв'язків-стовпців n , тобто n базисних стовпців матриці A .

Це означення опорного плану.

Висновок. Якщо цільова функція f обмежена на P , то існує кутова точка (опорний план), в якій досягається максимум. Тому для розв'язування задач лінійного програмування треба дослідити тільки вершини многогранника (опорні плани) задачі лінійного програмування.

4.4 Графічний метод розв'язування задачі лінійного програмування

В основі графічного методу лежить геометричний зміст задачі лінійного програмування. Застосовується цей метод у випадку , інакше важко будувати багатокутник розв'язків. Нехай задана задача лінійного програмування

;

Припустимо, що система , сумісна і багатокутник розв'язків задачі обмежений. Кожна з нерівностей , визначає півплощину з межею ,

Перетин цих півплощин визначає багатокутник .

Цільова функція при фіксованих значеннях є рівнянням прямої

Побудуємо і графік цільової функції при . Тоді задача - означає: знайти вершину багатокутника розв'язків, у якій пряма є опорною і при цьому набуває найбільшого (найменшого) розв'язку.

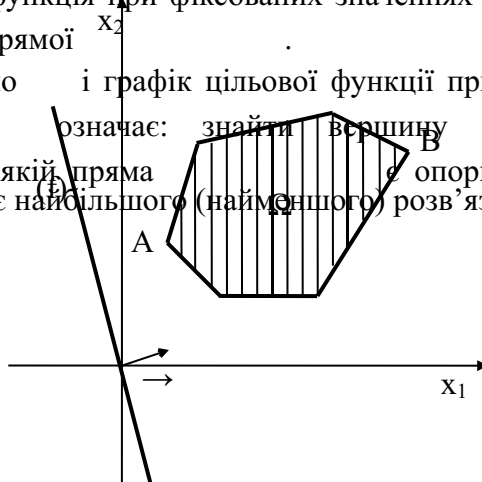


Рисунок 4.3

Відомо, що значення цільової функції зростає у напрямку вектора $\vec{n}$. Тому пряму l пересуваємо паралельно до себе у напрямку $\vec{n}$, поки вона не стане опорною для Ω . На рис.4.3 пряма l стає опорною до Ω в двох точках A і B . Враховуючи напрям вектора нормалі $\vec{n}$, одержимо, що максимум досягається в точці B , а мінімум – у точці A . Тут можливий випадок, коли мінімум (максимум) досягається у вершині або на стороні Ω .

Якщо Ω – необмежений, то можливі випадки:

- 1) пряма l постійно перетинає багатокутник Ω і ніколи не стає опорною до нього, тоді цільова функція необмежена знизу і зверху (рис.4.4):

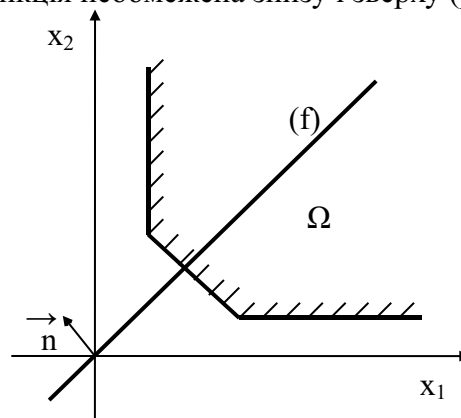


Рисунок 4.4

- 2) пряма стає опорною до :
 – обмежена зверху і необмежена знизу, – точка максимуму, мінімуму не існує (рис.4.5):

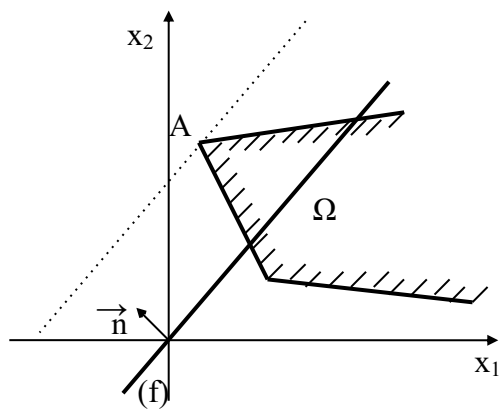


Рисунок 4.5

- 3) f - обмежена знизу і необмежена зверху, B - точка мінімуму, максимуму не існує (рис.4.6):

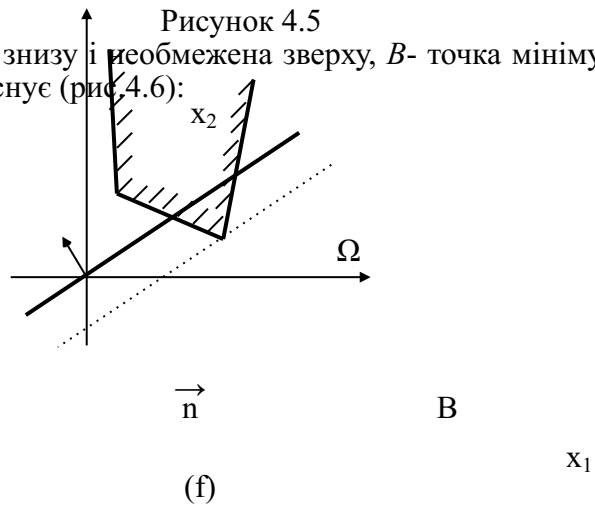


Рисунок 4.6

4) – обмежена знизу і зверху: на AC досягається максимум, на BD – мінімум (рис.4.7):

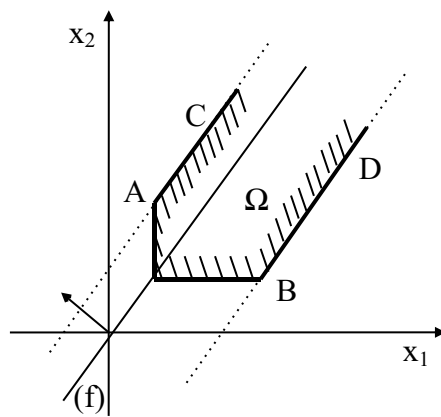


Рисунок 4.7

Якщо система – несумісна, то множина допустимих розв'язків порожня, тому задача оптимального розв'язку не має.

Приклад 4.1 Задача про використання сировини

;

, ;

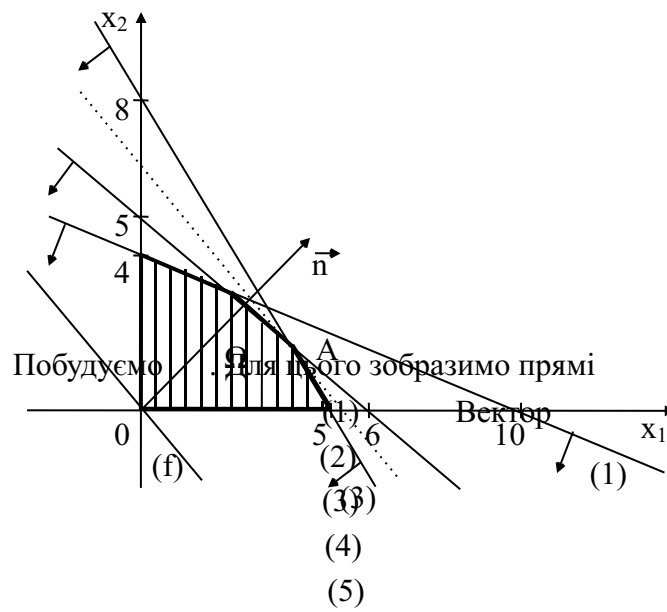


Рисунок 4.8

Точка A – точка максимуму. Знайдемо її координати. Оскільки A – точка перетину (2) і (3), то її координати задовольняють систему

$$(2)$$

$x_1 \geq 0$; $x_2 \geq 0$;
 $x_3 \geq 0$; $x_4 \geq 0$;
 $x_5 \geq 0$; $x_6 \geq 0$;

Отже, .

Відповідь: .

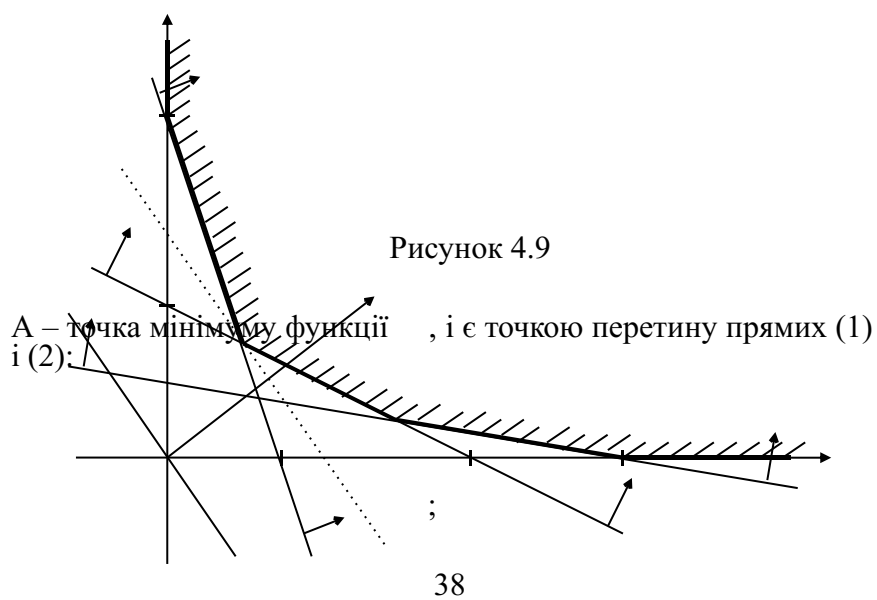
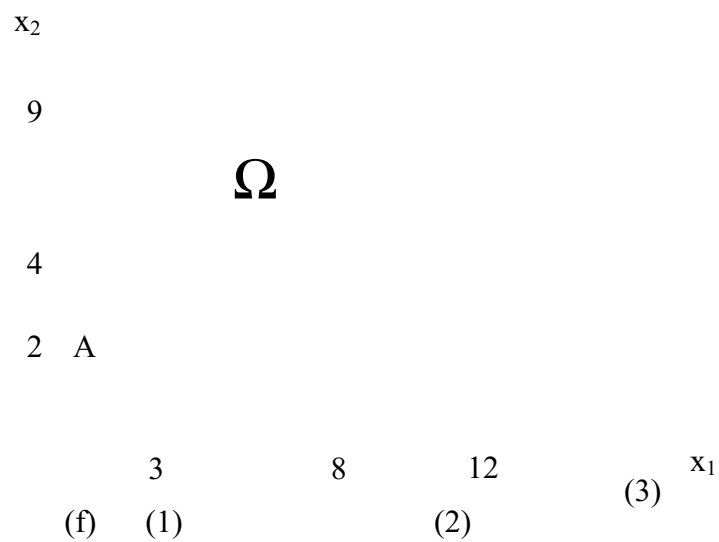
Приклад 4.2. Задача про дієту

$x_1 \geq 0$; $x_2 \geq 0$;

Межі : $x_1 \leq 10$; (1)
 $x_2 \leq 15$; (2)
 $x_3 \leq 20$; (3)
 $x_4 \leq 25$.

Вектор

Побудуємо многокутник розв'язків:



Отже,

Відповідь:

Зауваження. За допомогою графічного методу можна розв'язувати задачі лінійного програмування з n змінними і m обмеженнями-рівностями, якщо

Нехай рівняння системи обмежень лінійно незалежні. Тоді методом Жордана-Гаусса можна знайти m базисних змінних, які виражаються через $n-m$ вільні змінні. Оскільки $n-m=2$, то вільних змінних буде дві: x_1 і x_2 . Таким чином, всі базисні змінні виражаються через вільні. Тоді, підставивши замість базисних змінних у цільову функцію їх вирази через змінні x_1 і x_2 , і перейшовши до нерівностей із двома змінними, одержимо задачу з двома невідомими. Розв'язавши одержану задачу графічно, одержимо x_1 і x_2 . Таким чином, знайдемо розв'язок.

Приклад 4.3. Розв'язати задачу лінійного програмування

;

Тут x_1, x_2, x_3, x_4, x_5 – змінні.

Розв'яжемо систему обмежень. Нехай x_3, x_4, x_5 – базисні

			1	1	
			1	-1	

x_5

A

			-1	(1)	
			0	2	
			1	1	
			-1	1	
			0	2	
			2	-2	
			-1	1	
				-1	

1

0,5

1,5

x_4

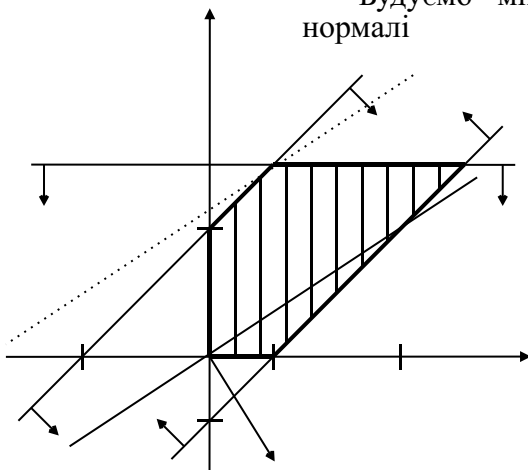
(f)

(2)

Підставимо у :

Система обмежень матиме вигляд

Маємо задачу з двома змінними x_1 і x_2 .
Будуємо многокутник розв'язків, лінію рівня, вектор нормалі



- (рис. 4.10)

Рисунок 4.10
З рисунка визначаємо: А – точка мінімуму .

$\frac{1}{2}$, $\frac{1}{2}$, $\frac{1}{2}$, $\frac{1}{2}$, $\frac{1}{2}$, $\frac{1}{2}$.

Відповідь: ; .

5 Симплексний метод розв'язування задач лінійного програмування

5.1 Обґрунтування симплекс-методу

Основним методом розв'язування задач лінійного програмування є розроблений у 1949 році американським математиком Д. Данцігом симплекс-метод. Він застосовується для задач у канонічній формі

$$z = c_1x_1 + c_2x_2 + \dots + c_nx_n \rightarrow \min$$

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n + x_{n+1} = b_1$$

$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n + x_{n+2} = b_2$$

$$\vdots$$

$$a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n + x_{n+m} = b_m$$

$$x_1, x_2, \dots, x_n, x_{n+1}, x_{n+2}, \dots, x_{n+m} \geq 0$$

;

де
 Нагадаємо, що з властивостей розв'язків задачі лінійного програмування випливає: оптимальний план задачі (1)-(3) знаходиться серед опорних планів (вершин многогранника розв'язків), тобто серед таких розв'язків системи (2)-(3), додатним координатам яких відповідають лінійно незалежні стовпці A_j матриці A . Симплексний метод організовує знаходження і перегляд опорних планів у певному порядку. Перехід від одного опорного плану до іншого здійснюється переведенням однієї з небазисних змінних у базисну. При цьому цільова функція не зменшується.

Нехай система (2) є системою у базисній формі. Це означає, що є m базисних змінних (наприклад $x_1, x_2, \dots, x_m$), кожна з яких входить лише в одне рівняння з коефіцієнтом 1 і всі числа , . Тобто система (2) має вигляд

Тоді загальний розв'язок системи (4) такий:

а базисний розв'язок буде ,
 або ,
 або .

Оскільки $\bar{b} \geq 0$, то це опорний план.
Підставимо (5) у цільову функцію (1):

Позначимо $x_1 = \bar{b} - \frac{1}{a_{11}}x_2$, $x_2 = x_2$, $x_3 = 0$;
 $x_4 = 0$, $x_5 = 0$, $x_6 = 0$.

Тоді
Вираз (8) називається зведеним виразом цільової функції,
а числа $\bar{c}_j$ - зведеними коефіцієнтами цільової функції.

Зауваження. З системи (5) випливає, що $x_1 \geq 0$,
 $x_2 \geq 0$. Справді, при $x_2 = 0$, $x_1 = \bar{b} \geq 0$, тому
 $x_1 \geq 0$, а тоді оцінки базисних змінних
дорівнюють нулеві.

Проаналізуємо зв'язок між величиною цільової функції і
вільними змінними.

Випадок 1. Нехай усі значення $\bar{c}_j$ при вільних змінних
у функції невід'ємні, $\bar{c}_j \geq 0$.

Тоді всяке збільшення вільних змінних
приводить до зменшення функції (8). Тому найбільше
значення функція буде мати при $x_2 = 0$, тобто у
допустимому базисному розв'язку (6). Тоді (6) є оптимальним
розв'язком.

Таким чином, умовою оптимальності є умова

Випадок 2. Серед оцінок вільних змінних є принаймні одна від'ємна і при цьому серед коефіцієнтів відповідного їй j -го стовпчика матриці немає додатних. У цьому випадку функція є необмеженою в області допустимих розв'язків.

Справді, нехай для деякої вільної змінної оцінка x_j і всі x_i (де $i \in I$). Розглянемо всі вільні змінні, крім x_j , які є рівними нулю і підставимо у (5) і (8). Одержимо

Оскільки $a_{ij} < 0$, то при збільшенні x_j функція f зростає. Разом з тим з (10) випливає, що усі змінні x_i залежні від x_j , залишаються невід'ємними при довільних як завгодно великих x_j (бо $a_{ij} < 0$).

Отже, функція f максимуму не має, оптимального плану не існує, хоч є безліч допустимих планів.

Випадок 3. Є вільна змінна x_s з від'ємною оцінкою x_s , у якої принаймні один коефіцієнт a_{is} є додатним: $a_{is} > 0$. Тоді можна знайти новий опорний план, для якого значення цільової функції є більшим, якщо розглядуваний опорний план не вироджений.

Справді, як видно з (11), при збільшенні x_s приводить до зростання цільової функції f . Збільшення припустиме доти, поки праві частини виразу (10) залишаються

невід'ємними при всіх x_i . Тобто для x_s $\theta = \min_{i \in I} \{ -a_{is} x_i / a_{is} \}$. Максимально можливе значення x_s при якому забезпечується невід'ємність змінних x_i в (10), визначається з умови

Для невиродженого опорного плану всі $\bar{c}_j \geq 0$, отже

. Таким чином, беручи x_{m+1} за базисну змінну, ми переведемо вільну змінну x_{m+1} у число базисних замість базисної змінної x_m . Для визначення значень решти базисних змінних $x_1, \dots, x_m, x_{m+2}, \dots, x_n$, виконаємо перетворення Жордана-Гаусса.

A_1	...	A_m	A_{m+1}	...	A_s	...	A_n	B
1	...	0		...		...		
...	...	...	...	...	...	...	...	...
0	...	0		...		...		
...	...	...	...	...	...	...	...	...
0	...	0		...		...		
...	...	...	...	...	...	...	...	...
0	...	1		...		...		
	...			...		...		

(14)

Зауваження. Якщо мінімум відношень досягається у кількох рядках, то нульових значень набудуть декілька відповідних змінних. Нова базисна змінна x_s вводиться замість будь-якої з них, а решта залишаються у базисі. Тоді одержаний план буде виродженим.

Покажемо, як змінюється цільова функція при переході до нового опорного плану (14). Для цього підставимо (14) в (8) або в (11):

Оскільки, то цільова функція зростає з

до (на величину).
Формула (15) нагадує формулу прямокутника жорданових виключень. Запишемо (8) у виді

Долучимо це рівняння до системи (4) як таке, що розв'язане відносно невідомої базисної змінної. При проведенні жорданового перетворення з провідним елементом одержимо перетворені елементи

Отже, виконавши жорданове перетворення системи (4), розширеної додаванням рядка зведеної цільової функції (16), одержимо новий допустимий базисний рядок (14), для якого значення цільової функції буде більшим.

Щоб вияснити, чи буде новий базисний розв'язок (14)

оптимальним, треба проаналізувати оцінки θ . Якщо маємо випадок 1, то знайдений план є оптимальним, якщо випадок 2, то оптимального плану не існує. Якщо випадок 3, то описаний процес продовжують. Оскільки кількість опорних планів не перевищує n , то процес є скінченний.

Якщо в процесі ітерацій появиться вироджений план, то перехід до наступного опорного плану відбувається без збільшення значення цільової функції.

Теоретично можливий випадок, коли послідовність вироджених опорних планів починає повторюватись – зациклюватись. У цьому випадку розроблені спеціальні прийоми. На практиці зациклованості майже не буває.

5.2 Алгоритм застосування симплекс-методу

Розглянемо задачу у базисній формі (1), (3), (4), де $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$, $x \in \mathbb{R}^n$, $x \geq 0$, $b \geq 0$.

Умови такої задачі зручно записувати у вигляді симплексної таблиці

		C_B				...			...	
						...			...	
1				1	0	...			...	
2				0	1	...			...	
						...			...	
				0	0	...			...	
$m+1$	Δ					...			...	

У m -му (оцінковому) рядку записують числа $c_j - C_B A_j$.

У стовпчику Δ (базис) записуються одиничні базисні

вектори, у стовпчику — коефіцієнти цільової функції, що відповідають векторам базису, у стовпчику B — значення базисних змінних, у стовпчиках $A_1, A_2, \dots, A_n$ — коефіцієнти a_{ij} системи обмежень. Над позначеннями $A_1, A_2, \dots, A_n$ записують коефіцієнти цільової функції при відповідних змінних.

Алгоритм розв'язування задачі лінійного програмування симплекс-методом

1. Заповнюємо симплексну таблицю, знаходимо оцінки
, значення .
 2. Переглядаємо знаки коефіцієнтів . Якщо всі ,
то задача розв'язана і базисний розв'язок
буде оптимальним, а . Якщо не всі ,
переходимо до п.3.
 3. Серед від'ємних значень знаходимо найбільше
по модулю число і відповідний йому стовпчик вибираємо за
провідний. Нехай це буде стовпчик з номером . Якщо в
провідному стовпчику всі елементи (), то
маємо випадок 2: цільова функція необмежена і
розв'язування закінчене. Якщо не всі , то перейти до п.
4.
 4. Для кожного додатного елемента -го
(провідного) стовпчика знаходимо відношення , вибираємо
найменше (тобто знаходимо симплексне відношення), і
називаємо рядок, де воно досягається, провідним. Нехай це
буде рядок з номером . Елемент на перетині провідного
рядка і провідного стовпця буде провідним.
 5. Виконуємо жорданове перетворення таблиці від
стовпця B до A_n , від 1-го рядка до (+1)-го з провідним
елементом і переходимо до кроку 2.
- Послідовність операцій 2-5 називається **ітерацією**
симплекс-методу.

Зауваження 1. У другій і далі таблицях і можна шукати за правилом прямокутника як і інші елементи.

Зауваження 2. Якщо на кроці 3 є декілька найбільших по модулю оцінок, то в базис включають той вектор, якому відповідає найбільше.

Точнішим є таке правило: якщо не виконується умова оптимальності, то в базис включають той вектор, якому відповідає $\min \Delta_j$, при $\Delta_j < 0$.

Якщо ж мінімальних оцінок декілька, то беруть вектор, якому відповідає максимальне.

Зауваження 3. Якщо в (1), то умовою оптимальності плану є $\Delta_j \leq 0, j=1, \dots, n$.

Приклад 5.1. Розв'язати задачу лінійного програмування

Складаємо симплекс-таблицю

		C_b		4	1	1	-3	0
1		1	6	2	-1	1	0	0
2		-3	8	1	-1	0	1	0
3		0	18	1	1	0	0	1
			-18	-5	1	0	0	0
1		4	3	1			0	0
2		-3	5	0			1	0

3		0	15	0			0	1
			-3	0			0	0
1		4	8	1	0		0	
2		-3	10	0	0		1	
3		1	10	0	1		0	
			12	0	0	2	0	1

Всі ≤ 0 , тому базисний розв'язок є оптимальний.

Відповідь:

Приклад 5.2. Розв'язати симплекс-методом задачу лінійного програмування

;

Зведемо задачу до канонічного вигляду

;

Складемо симплекс-таблицю

i				-1	1	3	0	0	0
1		0	1	2	-1	1	1	0	0
2		0	2	-4	2	-1	0	1	0
3		0	5	3	0	1	0	0	1
$m+1$	Δ		0	1	-1	-3	0	0	0
1		3	1	2	-1	1	1	0	0
2		0	3	-2	1	0	1	1	0
3		0	4	1	1	0	-1	0	1
			3	7	-4	0	3	0	0
1		3	4	0	0	1	2	1	0
2		1	3	-2	1	0	1	1	0
3		0	1	3	0	0	-2	-1	1
			15	-1	0	0	7	4	0
1		3	4	0	0	1	2	1	0
2		1	$\frac{11}{3}$	0	1	0			
3		-1	$\frac{1}{3}$	1	0	0			
				0	0	0			

Зауваження 4. Як відзначалося, при переході від виродженого опорного плану до нового опорного плану цільова функція не збільшується. Можливий випадок, коли новий опорний план також вироджений і т.д., тоді можливе зациклення (коли вироджені плани повторюються). Для усунення зациклення і покращення розв'язку використовують таке правило. Якщо на кроці 4 алгоритму симплекс-методу

досягається у декількох рядках, то треба вибрати той рядок, для якого буде найменшим відношення елементів наступного стовпчика до провідного. Якщо при цьому відношення виявляється рівним, то беруть відношення іншого стовпчика до провідного і т.д., поки не виявиться однозначно провідний елемент.

Приклад 5.3. Розв'язати задачу лінійного програмування симплекс-методом

;

Задача записана в канонічному базисному вигляді.

Складаємо симплексну таблицю

i				2	4	0	0	0	0
1		0	6	-2	1	1	0	0	0
2		0	9	-1		0	1	0	0

3		0	30	-1	5	0	0	1	0
4		0	12	1	1	0	0	0	1
			0	-2	-4	0	0	0	0
1		4	6	-2	1	1	0	0	0
2		0	0	2	0		1	0	0
3		0	0	9	0	-5	0	1	0
4		0	6	3	0	-1	0	0	1
			24	-10	0	4	0	0	0
1		4	6	0	1		1	0	0
2		2	0	1	0			0	0
3		0	0	0	0			1	0
4		0	6	0	0			0	1
	Δ		24	0	0		5	0	0
1		4	6	0	1	0			0
2		2	0	1	0	0			0
3		0	0	0	0	1			0
4		0	6	0	0	0			1
			24	0	0	0		2	0
1		4	7	0	1	0	0		

2		2	5	1	0	0	0		
3		0	9	0	0	1	0		
4		0		0	0	0	1		
			38	0	0	0	0		

Відповідь: — невикористаний оптимальний план,

6 Метод штучного базису (М – метод) розв’язування задач лінійного програмування

У попередньому розділі доведено, що задачу лінійного програмування можна розв’язати симплекс-методом, якщо вона записана у канонічній формі, всі і система обмежень має одиничну матрицю порядку . Тоді існує початковий опорний план, виходячи з якого за допомогою симплекс-методу знаходиться оптимальний план.

Знаходження початкового опорного плану складає окрему задачу. Деколи побудувати опорний план просто. Наприклад, коли система обмежень має вигляд , то введенням додаткових змінних обмеження зводяться до потрібного вигляду і початковим опорним

планом буде план

Знайти початковий опорний план можна методом

Жордана-Гаусса, але тоді треба перебирати кілька допустимих розв'язків.

Багато задач лінійного програмування не містять одиничної матриці. У цьому випадку для розв'язування задач застосовується метод штучного базису, або — методу.

Розглянемо задачу у канонічній формі

;

де система обмежень не містить одиничної матриці.

До кожної рівності з (2) введемо по одній змінній і розглянемо розширену задачу (— задачу).

;

Вважається, що число є додатним і досить великим.

Тут називаються штучними змінними.

Вектор-стовпчики є одиничними і утворюють штучний базис.

Система (5)-(6) має базисний розв'язок , який можна взяти за початковий опорний план задачі (4)-(6).

Далі задачу (4)-(6) розв'язують симплекс-методом. Можливі випадки:

1. Знайдено оптимальний розв'язок задачі (4)-(6) , у якому всі штучні змінні дорівнюють нулю. Тоді задача (1)-(3) має

оптимальний розв'язок .

Оскільки і відрізняються останніми нульовими координатами і – оптимальний план - , то – допустимий розв'язок.

Доведемо, що – оптимальний.

Нехай це не так. Тоді існує план такий, що , а тому існує план розширеної задачі такий, що ,

тобто , що суперечить тому, що – оптимальний план.

2. Оптимальний розв'язок містить хоч одну додатну штучну змінну . Тоді можна довести, що початкова задача (1)-(3) не має допустимих розв'язків, тому нерозв'язна.

3. M -задача не має розв'язків. Тоді задача - також нерозв'язна (функція необмежена).

Якщо число M відоме, то задачу (4)-(6) розв'язують звичайним симплекс-методом.

Якщо число M невідоме, то симплекс-метод модернізують: в симплексну таблицю вводять -й рядок. Коли обчислюють оцінки , то вони є лінійними функціями від , тобто мають вигляд . Числа записують в

-й рядок, а числа – в -й. Ітерацію симплекс-методу проводять по -у рядку за звичайними правилами. Можливі випадки:

- 1) всі штучні змінні будуть виведені з базису. У цьому випадку далі перерахунок симплекс-таблиць проводять по $(m+1)$ -му рядку, причому обчислення в стовпцях із штучними змінними можна не проводити.
- 2) всі числа в -му рядку у стовпцях стануть додатні, хоч не всі штучні змінні виведені з базису. У цьому випадку, якщо елемент в -му рядку стовпчика від'ємний, то вихідна задача - не має розв'язку. Якщо це число дорівнює нулеві, то знайдений опорний план вихідної задачі є виродженим і базис містить принаймні один вектор базису. Тоді знову переходять до -го рядка.

Висновок: При розв'язуванні канонічної задачі лінійного програмування можливі такі випадки:

- 1) матриця системи обмежень містить одиничну матрицю порядку . Тоді задачу розв'язують звичайним симплекс-методом;
- 2) матриця містить різних одиничних стовпчиків. Тоді вводять штучних змінних і розв'язують -задачу;
- 3) матриця не містить одиничних векторів. Треба розв'язувати -задачу з штучними змінними.

Приклад 6.1. Розв'язати задачу лінійного програмування методом штучного базису

;

Випишемо матрицю системи:
 Система обмежень не містить жодного одиничного вектора. Тому введемо дві штучні змінні , і розглянемо M -задачу:

;

Розв'яжемо M -задачу

i			B	5	3	4	-1	$-M$	$-M$
1	A_5	$-M$	3	1	3	2	2	1	0
2	A_6	$-M$	3	2	2	1	1	0	1
$m+1$			0	-5	-3	-4	1	0	0
$m+2$			-6	-3	-5	-3	-3	0	0
1	A_2	3	1		1				0
2	A_6	$-M$	1		0				1
$m+1$			3	-4	0	-2	3		0
$m+2$			-1		0				0
1	A_2	3		0	1				
2	A_1	5		1	0				

$m + 1$			6	0	0	-3	2		
$m + 2$			0	0	0	0	0		
1	A_3	4	1	0		1	1		
2	A_I	5	1	1		0	0		
$m + 1$			9	0	4	0	5		

Оскільки всі $\Delta_j \geq 0$, то розв'язок є оптимальний для -задачі.

Відповідь:

Приклад 6.2. Розв'язати задачу лінійного програмування

;

Зведемо задачу до канонічного виду:

;

,

Матриця має один одиничний вектор, тому треба ввести лише дві штучні змінні і розглянути -задачу:

				-2	1	-3	7	0		
1			6	1	2	0	3	0	1	0
2			10	4	-1	1	-1	0	0	1
3		0	14	-1	3	2	1	1	0	0
			0	2	-1	3	-7	0	0	0
			-16	-5	-1	-1	-2	0	0	0
1				0				0	1	
2		-2		1				0	0	
3		0		0				1	0	
			-5	0				0	0	
				0				0	0	
1		7		0			1	0		
2		-2		1			0	0		
3		0		0			0	1		
			2	0	4	2	0	0		

--	--	--	--	--	--	--	--	--	--	--	--

Всі , тому , .

7 Двоїсті задачі лінійного програмування

З кожною задачею лінійного програмування тісно пов'язана інша задача, така, що при розв'язуванні однієї із них одночасно розв'язується і друга. Ці задачі називаються *двоїстими (спряженими)* задачами.

Розглянемо стандартну задачу

;

або у матричній формі ,

Введемо ; розгляд m-вимірний вектор-рядок

;

або у матричній формі ,

Задача (3) називається **двоїстою** (спряженою) до задачі (1). Відповідно задача (1) називається **прямою** у відношенні

до задачі (3).

Часто двоїсті задачі мають конкретний економічний зміст. Побудуємо пару двоїстих задач.

Нехай прямою є задача використання сировини: підприємство має видів сировини, з яких випускає видів продукції. На одиницю -того виду продукції витрачається одиниць -того виду сировини. Запас -того виду сировини складає одиниць, . Прибуток від реалізації одиниці -того виду продукції дорівнює одиниць, . Треба знайти план випуску продукції, при якому прибуток є максимальним. Математичною моделлю цієї задачі є задача (1).

7.1 Економічна інтерпретація пари двоїстих задач

Нехай задано величини , , , які мають той же зміст, що і у вихідній задачі про використання сировини. Нехай додатково введено ще одну умову: сировину можна направити або на виготовлення продукції або на продаж іншому підприємству. Треба знайти, яку мінімальну ціну треба встановити за одиницю кожного -того виду сировини () при умові, що прибуток від реалізації всіх її запасів повинен бути не меншим за прибуток від реалізації продукції, яка може бути вироблена з цієї сировини.

Визначимо через шукану ціну одиниці сировини -того виду. Прибуток, який одержимо від продажу сировини, необхідної для виготовлення одиниці продукції -того виду, дорівнює

Для того, щоб продаж сировини був не менш вигідним, ніж реалізація виробленої з неї продукції, повинна виконуватись нерівність:

З порівняння задач (1) і (3) випливають **правила побудови задачі, двоїстої до задачі (1).**

1. Змінних в задачі (3) стільки, скільки обмежень в задачі (1).

2. Задача максимізації переходить у задачу мінімізації.

3. Матриця умов задачі (3) - це транспонована матриця умов задачі (1).

4. Коефіцієнти при невідомих у цільовій функції задачі (3)- це вільні члени з обмежень задачі (1). Обмеження у задачі (3) - це коефіцієнти у цільовій функції прямої задачі (1).

5. Якщо змінна вихідної задачі (1) є невід'ємною, то - та умова в системі обмежень задачі (3) є нерівністю типу .

Якщо в (1) змінна довільного знаку, то -те обмеження в (3) є рівністю. Якщо -те співвідношення в (1) є рівністю , то , якщо i -те співвідношення в (1) є рівністю- змінна може бути довільного знаку.

Описані правила можна подати такою схемою (випадок, коли цільова функція досліджується на максимум):

;

на умова на знак
не накладається;

;

умова на
не накладається.

У вихідній задачі умови можуть бути інші. Наприклад , обмеження , . Тоді такі задачі треба

звести до одного з таких типів.

Пряма задача		Двоїста задача	
1	;	;	;
	;	;	;
2	;	;	;
	;	;	;
3	;	;	;
	;	;	;
4	;	;	;
	;	;	;

де - матриця, транспонована до даної.

Приклад 7.1 Записати задачі, двоїсті до заданих

а)

б)

Розв'язання.

а) Зведемо задачу до вигляду (2)

;

Двоїста: ;

б) Двоїста задача: ;

.
Замінімо: ;
;

Тоді , , , .
;

У прикладі б) задачу до основних виразів не зводили, а використали загальну постановку несиметричної двоїстої пари задач.

Якщо пряма задача має вигляд

;

то двоїста буде такою:

Зауваження. Задача, двоїста до двоїстої, є прямою задачею.

Висновок. Потрібно пам'ятати основне:

- 1) умова змінюється на ;
- 2) максимуму цільової функції відповідає знак у обмеженнях, мінімуму –

відповідає знак $\leq$ у обмеженнях, при умові, що змінні невід'ємні.

7.2 Зв'язок між розв'язністю прямої і спряженої задачі

Вивчимо зв'язок між прямою і спряженою задачею. Сформулюємо основні властивості стосовно пари симетричних двоїстих задач (P) і (D) . Означимо множини допустимих розв'язків задач P і D відповідно

Лема 1. Для довільних допустимих планів $x \in P$, $y \in D$,
прямої і спряженої задачі правильна нерівність

Оскільки $c^T x \leq b^T y$, тоді $c^T x \leq b^T y$.
Аналогічно, з того, що $b^T y \leq c^T x$, випливає $b^T y \leq c^T x$.
Тому $c^T x = b^T y$.

Доведення для несиметричної пари задач проводиться аналогічно.

Нерівність (5) називається *основною нерівністю теорії двоїстості*. Її економічний зміст полягає в тому, що при довільному допустимому плані виробництва загальна вартість всього виробленого не перевищує сумарної оцінки ресурсів, що відповідає будь-якому допустимому вектору оцінок y .

Лема 2. Якщо для деяких допустимих планів $x \in P$, $y \in D$,
пари двоїстих задач виконується рівність $c^T x = b^T y$,

то вектори x і y є оптимальними планами відповідно прямої і спряженої задачі.

Нагадаємо, що вектор x оптимальний тоді і тільки тоді, коли для довільного допустимого вектора y правильна нерівність $c^T x \leq b^T y$.

Доведення. Нехай x – допустимий розв'язок прямої задачі. Тоді за лемою 1 і рівністю $c^T x = b^T y$ маємо:

тобто x^* -оптимальний. Аналогічно доводять оптимальність

Лема 2 дає критерій оптимальності для задач лінійного програмування. Згідно з цією лемою, якщо серед допустимих розв'язків знайдуться вектори x^* і y^* , що задовольняють (6), то вони будуть оптимальними розв'язками відповідних задач. З економічної точки зору це означає, що плани виробництва і оцінки ресурсів є оптимальними, якщо вартість всієї продукції і сумарна оцінка ресурсів збігаються.

Перша теорема двоїстості (теорема існування оптимального розв'язку). Якщо пряма і спряжена задача мають допустимі розв'язки, то вони мають оптимальні розв'язки, причому значення їх цільових функцій рівні.

Іншими словами, якщо одна з пари двоїстих задач має оптимальний розв'язок, то розв'язна і друга задача, причому значення цільових функцій рівні. Якщо x^* або y^* необмежена, то і інша задача не має розв'язку.

Друга теорема двоїстості. Допустимі розв'язки двоїстих задач x^* і y^* , оптимальні тоді і тільки тоді, коли виконуються умови

$$c_j - \sum_{i=1}^m a_{ij} y_i = 0, \quad j = 1, \dots, n; \quad (7)$$

або в розгорнутому виді

;

Умови (7) називаються умовами доповнюючої не жорсткості.

Доведення.

Достатність

З доведення леми 1: для довільних α, β ,

Нехай умови (7) виконуються. Тоді в (8) обидві нерівності стануть рівностями, тому x^* і y^* за лемою 2 – оптимальні розв’язки. (8)

Необхідність

Нехай одна з умов не виконується, наприклад $\alpha_i < 0$. Оскільки $x_i \geq 0$, то $\alpha_i x_i < 0$, а тому з (8) $\alpha_i x_i < \beta_i$. Звідси випливає, що принаймні один з розв’язків або не є оптимальним для відповідної задачі.

Аналогічно розглядається невиконання умови $\beta_i < 0$.

Таким чином, оптимальні розв’язки пари двоїстих задач володіють властивістю ортогональності: якщо $x_i > 0$ -та компонента оптимального вектора x^* додатна, то $y_i = 0$ -те обмеження спряженої задачі повинно виконуватись як рівність. Аналогічно якщо додатною є y_i -та компонента оптимального вектора y^* , то $x_i = 0$ -те обмеження спряженої задачі повинно виконуватись як рівність.

З теореми двоїстості випливають наслідки:

Наслідок 1. Якщо одна з двоїстих задач має оптимальний розв’язок, то і друга має оптимальний розв’язок.

Наслідок 2. Якщо одна з двоїстих задач має допустимий розв’язок, а друга ні, то цільова функція прямої задачі необмежена.

Зв’язок між розв’язками двоїстих задач виявляється в тому, що при розв’язуванні одної одночасно розв’язується і друга.

Справді, легко бачити, що симплекс-таблиця умовами вихідної задачі включає і умови спряженої задачі. Так, рядки цієї таблиці містять ліві частини обмежень вихідної задачі, стовпці – ліві частини обмежень спряженої задачі. Елементи стовпчика b є коефіцієнтами вектора обмежень прямої задачі і одночасно оцінками плану спряженої задачі. Елементи

-го рядка дають оцінки плану вихідної задачі і одночасно коефіцієнти вектора обмежень спряженої задачі. Тому при розв'язуванні двоїстої задачі можна не будувати спеціальну симплекс-таблицю, а розв'язувати її, використовуючи таблицю вихідної задачі.

Оптимальні розв'язки двоїстих задач знаходяться одночасно. Компоненти оптимального розв'язку двоїстої задачі знаходяться в клітинках -го рядка останньої таблиці, які відповідають початковому одиничному базису прямої задачі і дорівнюють сумі відповідного елемента цього

рядка і .

Приклад 7.2 Розв'язати задачу

;

;

і двоїсту до неї.

Запишемо спряжену задачу

;

Розв'яжемо вихідну задачу

				0	1	0	-1	-3	0
1		0	1	1	2	0	-1	1	0
2		0	2	0	-4	1	2	-1	0
3		0	5	0	3	0	0	1	1

			0	0	-1	0	1	3	0
1		1			1	0			0
2		0	4	2	0	1	0	1	0
3		0			0	0			1
					0	0			0

Отже, оптимальний план прямої задачі

Знайдемо оптимальний план двоїстої задачі. Його координати одержимо з -го рядка останньої ітерації. Щоб знайти -ту координату, треба знайти стовпчик, у якому в початковому базисі знаходився вектор з одиничкою у -ому рядку. З вибраного стовпця беремо оцінку і додаємо до неї відповідне значення коефіцієнта цільової функції :

$$, , .$$

Отже, , .

8 Двоїстий симплекс-метод розв'язування задач лінійного програмування

Симплекс-метод застосовується для розв'язування задач з невід'ємними правими частинами обмежень і довільними за знаком оцінками . Деколи легше знайти базис, який задовольняє ознаку оптимальності (всі), але не задовольняє умови . Варіант симплекс-методу, який

застосовується для розв'язування таких задач, називається *двоїстим симплекс-методом*. З його допомогою розв'язуються задачі лінійного програмування

$$\begin{aligned} & \text{;} \\ & \text{;} \\ & , \end{aligned}$$

де матриця містить одиничний базис і всі , .

При цьому умова , не вимагається. Таку задачу називають задачею у двоїстій базисній формі.

Звичайний симплекс-метод, який застосовується до задач в базисній формі, приводить до послідовності еквівалентних задач із зростаючим значенням цільової функції і невід'ємними значеннями , тобто кожний базисний розв'язок є допустимим.

Двоїстий симплекс-метод, який застосовується до задачі у двоїстій базисній формі, призводить до послідовності задач із спадним значенням цільової функції, невід'ємними коефіцієнтами і значеннями довільного знаку. Перетворення задач проводиться доти, поки не буде встановлено, що вихідна задача не має допустимого розв'язку, або не буде одержаний допустимий розв'язок (всі), який є одночасно оптимальним.

Тому звичайний симплекс-метод називають методом послідовного покращення планів, а двоїстий – методом послідовного уточнення планів.

Розглянемо задачу у двоїстій базисній формі. Можливі такі випадки:

Випадок 1. Всі коефіцієнти вектора обмежень невід'ємні: , . Оскільки всі , то одержано доступний оптимальний план.

Випадок 2. Є рядок , такий, що : всі , . Тоді система обмежень несумісна, тоді задача нерозв'язна.

Випадок 3. Є рядок , такий, що і хоч для одного , . Нехай таке, що і

Тоді жорданове перетворення з провідним елементом приводить до еквівалентної задачі, у якій, по-перше, оцінки , а, по-друге, значення цільової функції , і якщо , то .

Алгоритм двоїстого симплекс-методу

1. Перевіряємо знаки . Якщо всі , то в стовпчику записані базисний розв'язок і значення цільової функції, які дають оптимальний розв'язок вихідної задачі. Якщо не всі , перейти до кроку 2.
 2. Серед від'ємних вибираємо найбільше по модулю число і -тий рядок називається провідним.
 3. У провідному рядку перевіряємо знаки коефіцієнтів . Якщо всі , то задача розв'язку не має і нема жодного допустимого розв'язку. Якщо знайдеться , перейти до кроку 4.
 4. Серед від'ємних коефіцієнтів провідного рядка вибираємо елемент , для якого і називаємо провідним.
 5. Виконуємо жорданове перетворення з провідним елементом і переходимо до кроку 1.
- У результаті методу оптимальний план прямої задачі додається списком базисних змінних і їх значеннями у

стовпчику . Для двоїстої задачі оптимальний план має значення в рядку, в стовпчиках, які відповідають базисним стовпчикам вихідної задачі.

Приклад 8.1 . Розв'язати дану і двоїсту до даної задачі

;

Запишемо задачу у канонічному вигляді, у якому система обмежень має одиничний базис.

,

Розв'яжемо задачу двоїстим симплекс-методом

				1	1	2	0	0
1		2	8	1	1	1	0	0
2		0	-4	-1	1	0	1	0
3		0	-6	-1	-2	0	0	1
			16	1	1	0	0	0
1		2	5		0	1	0	
2		0	-7		0	0	1	
3		1	3		1	0	0	
			13		0	0	0	

1		2		0	0	1	
2		1		1	0	0	
3		1		0	1	0	
				0	0	0	

Запишемо двоїсту задачу

;

Запишемо розв'язок двоїстої з останньої симплекс-таблиці:

;

Відповідь:

;

Двоїстий симплекс-метод зручно використовується тоді, коли задача має одиничний базис, але не є у базисній, або у двоїстій базисній формі, тобто існують від'ємні i .

Розв'язання такої задачі складається з двох етапів і спочатку за допомогою двоїстого симплекс-методу виключаються всі від'ємні, потім оптимальний план

знаходиться симплекс-методом. Треба лише на першому етапі замінити крок 4 на 4/:

4/. Серед від'ємних коефіцієнтів вибираємо елемент , для якого виконується умова

Приклад 8.2

;

;

Запишемо задачу у базисній формі:

Двоїста задача:

;

Далі розв'яжемо пряму задачу двоїтим симплекс-методом

				8	2	-5	0	0	0
1		0	-6	1	-2	-1	1	0	0
2		0	-3	-1	2	-2	0	1	0

3		0	2	2	1	-1	0	0	1
			0	-8	-2	5	0	0	0
1		-5	6	-1	2	1	-1	0	0
2		0	9	-3	6	0	-2	1	0
3		0	8	1	3	0	-1	0	1
			-30	-3	-12	0	5	0	0
1		-5	3	0	0	1			0
2		2			1	0			0
3		0			0	0	0		1
			-12	-9	0	0	1	2	0
1		-5	3	0	0	1			0
2		2		0	1	0			
3		8		1	0	0	0		
				0	0	0	1		

Відповідь: ,

,

9 Цілочислове програмування

Існує доволі широкий клас задач математичного програмування, в економіко-математичних моделях яких одна, декілька або всі змінні мають набувати цілих значень.

Розглянемо задачу цілочислового програмування:

;

– ціле,

Для розв'язування цілочислових задач використовують метод відсікання або, як його ще називають, метод Гоморі.

Цілою частиною числа називають найбільше ціле число, яке не перевищує і позначають символом .

Наприклад, ; ;
Дробовою частиною числа називають різницю між

і його цілою частиною і позначають символом .

Наприклад, ; ; .

Очевидно, що , причому , тоді і тільки тоді, коли – ціле.

У **методі Гоморі** починають розв'язувати (1)–(4) за допомогою симплекс-методу. Якщо при цьому дістаємо оптимальний розв'язок, координати якого є цілими числами, то розв'язування задачі закінчено. Якщо ж необхідної цілочисловості немає, то будується відсікання, згідно з яким відсікається знайдений нецілочисловий результат, але в області залишаються усі допустимі точки з цілими координатами.

Нехай в оптимальному плані (1)–(4) є дробові координати.

Виберемо ту з них, дробова частина якої найбільша. Нехай це буде координата, що відповідає x_k -ому рядку таблиці, в якому одержали оптимальний, але не цілочисловий результат.

За допомогою коефіцієнтів цього рядка будемо відсікання

яке із введенням додаткової змінної x_{k+1} набуває вигляду

Додаємо одержане обмеження до останньої симплекс-таблиці, розв'язуємо поставлену тепер задачу за допомогою двоїстого симплекс-методу: беремо x_{k+1} змінну за базисну і для неї вводимо додатковий стовпчик.

Якщо в результаті одержаний оптимальний розв'язок ще містить нецілочислові координати, то будемо нові додаткові обмеження по вище описаній схемі для змінної x_{k+1} і процес обчислень продовжуємо до тих пір, доки всі координати оптимального розв'язку не стануть цілочисловими або задача не матиме розв'язку. Останнє має місце тоді, коли в рядку з дробовим усі цілі.

Приклад 9.1

За допомогою методу відсікань Гоморі знайти повністю цілочисловий оптимальний розв'язок задачі

$$\begin{aligned} & \max z = 2x_1 + 3x_2 + 4x_3 + 5x_4 + 6x_5 + 7x_6 + 8x_7 + 9x_8 + 10x_9 + 11x_{10} + 12x_{11} + 13x_{12} + 14x_{13} + 15x_{14} + 16x_{15} + 17x_{16} + 18x_{17} + 19x_{18} + 20x_{19} + 21x_{20} + 22x_{21} + 23x_{22} + 24x_{23} + 25x_{24} + 26x_{25} + 27x_{26} + 28x_{27} + 29x_{28} + 30x_{29} + 31x_{30} + 32x_{31} + 33x_{32} + 34x_{33} + 35x_{34} + 36x_{35} + 37x_{36} + 38x_{37} + 39x_{38} + 40x_{39} + 41x_{40} + 42x_{41} + 43x_{42} + 44x_{43} + 45x_{44} + 46x_{45} + 47x_{46} + 48x_{47} + 49x_{48} + 50x_{49} + 51x_{50} + 52x_{51} + 53x_{52} + 54x_{53} + 55x_{54} + 56x_{55} + 57x_{56} + 58x_{57} + 59x_{58} + 60x_{59} + 61x_{60} + 62x_{61} + 63x_{62} + 64x_{63} + 65x_{64} + 66x_{65} + 67x_{66} + 68x_{67} + 69x_{68} + 70x_{69} + 71x_{70} + 72x_{71} + 73x_{72} + 74x_{73} + 75x_{74} + 76x_{75} + 77x_{76} + 78x_{77} + 79x_{78} + 80x_{79} + 81x_{80} + 82x_{81} + 83x_{82} + 84x_{83} + 85x_{84} + 86x_{85} + 87x_{86} + 88x_{87} + 89x_{88} + 90x_{89} + 91x_{90} + 92x_{91} + 93x_{92} + 94x_{93} + 95x_{94} + 96x_{95} + 97x_{96} + 98x_{97} + 99x_{98} + 100x_{99} + 101x_{100} + 102x_{101} + 103x_{102} + 104x_{103} + 105x_{104} + 106x_{105} + 107x_{106} + 108x_{107} + 109x_{108} + 110x_{109} + 111x_{110} + 112x_{111} + 113x_{112} + 114x_{113} + 115x_{114} + 116x_{115} + 117x_{116} + 118x_{117} + 119x_{118} + 120x_{119} + 121x_{120} + 122x_{121} + 123x_{122} + 124x_{123} + 125x_{124} + 126x_{125} + 127x_{126} + 128x_{127} + 129x_{128} + 130x_{129} + 131x_{130} + 132x_{131} + 133x_{132} + 134x_{133} + 135x_{134} + 136x_{135} + 137x_{136} + 138x_{137} + 139x_{138} + 140x_{139} + 141x_{140} + 142x_{141} + 143x_{142} + 144x_{143} + 145x_{144} + 146x_{145} + 147x_{146} + 148x_{147} + 149x_{148} + 150x_{149} + 151x_{150} + 152x_{151} + 153x_{152} + 154x_{153} + 155x_{154} + 156x_{155} + 157x_{156} + 158x_{157} + 159x_{158} + 160x_{159} + 161x_{160} + 162x_{161} + 163x_{162} + 164x_{163} + 165x_{164} + 166x_{165} + 167x_{166} + 168x_{167} + 169x_{168} + 170x_{169} + 171x_{170} + 172x_{171} + 173x_{172} + 174x_{173} + 175x_{174} + 176x_{175} + 177x_{176} + 178x_{177} + 179x_{178} + 180x_{179} + 181x_{180} + 182x_{181} + 183x_{182} + 184x_{183} + 185x_{184} + 186x_{185} + 187x_{186} + 188x_{187} + 189x_{188} + 190x_{189} + 191x_{190} + 192x_{191} + 193x_{192} + 194x_{193} + 195x_{194} + 196x_{195} + 197x_{196} + 198x_{197} + 199x_{198} + 200x_{199} + 201x_{200} + 202x_{201} + 203x_{202} + 204x_{203} + 205x_{204} + 206x_{205} + 207x_{206} + 208x_{207} + 209x_{208} + 210x_{209} + 211x_{210} + 212x_{211} + 213x_{212} + 214x_{213} + 215x_{214} + 216x_{215} + 217x_{216} + 218x_{217} + 219x_{218} + 220x_{219} + 221x_{220} + 222x_{221} + 223x_{222} + 224x_{223} + 225x_{224} + 226x_{225} + 227x_{226} + 228x_{227} + 229x_{228} + 230x_{229} + 231x_{230} + 232x_{231} + 233x_{232} + 234x_{233} + 235x_{234} + 236x_{235} + 237x_{236} + 238x_{237} + 239x_{238} + 240x_{239} + 241x_{240} + 242x_{241} + 243x_{242} + 244x_{243} + 245x_{244} + 246x_{245} + 247x_{246} + 248x_{247} + 249x_{248} + 250x_{249} + 251x_{250} + 252x_{251} + 253x_{252} + 254x_{253} + 255x_{254} + 256x_{255} + 257x_{256} + 258x_{257} + 259x_{258} + 260x_{259} + 261x_{260} + 262x_{261} + 263x_{262} + 264x_{263} + 265x_{264} + 266x_{265} + 267x_{266} + 268x_{267} + 269x_{268} + 270x_{269} + 271x_{270} + 272x_{271} + 273x_{272} + 274x_{273} + 275x_{274} + 276x_{275} + 277x_{276} + 278x_{277} + 279x_{278} + 280x_{279} + 281x_{280} + 282x_{281} + 283x_{282} + 284x_{283} + 285x_{284} + 286x_{285} + 287x_{286} + 288x_{287} + 289x_{288} + 290x_{289} + 291x_{290} + 292x_{291} + 293x_{292} + 294x_{293} + 295x_{294} + 296x_{295} + 297x_{296} + 298x_{297} + 299x_{298} + 300x_{299} + 301x_{300} + 302x_{301} + 303x_{302} + 304x_{303} + 305x_{304} + 306x_{305} + 307x_{306} + 308x_{307} + 309x_{308} + 310x_{309} + 311x_{310} + 312x_{311} + 313x_{312} + 314x_{313} + 315x_{314} + 316x_{315} + 317x_{316} + 318x_{317} + 319x_{318} + 320x_{319} + 321x_{320} + 322x_{321} + 323x_{322} + 324x_{323} + 325x_{324} + 326x_{325} + 327x_{326} + 328x_{327} + 329x_{328} + 330x_{329} + 331x_{330} + 332x_{331} + 333x_{332} + 334x_{333} + 335x_{334} + 336x_{335} + 337x_{336} + 338x_{337} + 339x_{338} + 340x_{339} + 341x_{340} + 342x_{341} + 343x_{342} + 344x_{343} + 345x_{344} + 346x_{345} + 347x_{346} + 348x_{347} + 349x_{348} + 350x_{349} + 351x_{350} + 352x_{351} + 353x_{352} + 354x_{353} + 355x_{354} + 356x_{355} + 357x_{356} + 358x_{357} + 359x_{358} + 360x_{359} + 361x_{360} + 362x_{361} + 363x_{362} + 364x_{363} + 365x_{364} + 366x_{365} + 367x_{366} + 368x_{367} + 369x_{368} + 370x_{369} + 371x_{370} + 372x_{371} + 373x_{372} + 374x_{373} + 375x_{374} + 376x_{375} + 377x_{376} + 378x_{377} + 379x_{378} + 380x_{379} + 381x_{380} + 382x_{381} + 383x_{382} + 384x_{383} + 385x_{384} + 386x_{385} + 387x_{386} + 388x_{387} + 389x_{388} + 390x_{389} + 391x_{390} + 392x_{391} + 393x_{392} + 394x_{393} + 395x_{394} + 396x_{395} + 397x_{396} + 398x_{397} + 399x_{398} + 400x_{399} + 401x_{400} + 402x_{401} + 403x_{402} + 404x_{403} + 405x_{404} + 406x_{405} + 407x_{406} + 408x_{407} + 409x_{408} + 410x_{409} + 411x_{410} + 412x_{411} + 413x_{412} + 414x_{413} + 415x_{414} + 416x_{415} + 417x_{416} + 418x_{417} + 419x_{418} + 420x_{419} + 421x_{420} + 422x_{421} + 423x_{422} + 424x_{423} + 425x_{424} + 426x_{425} + 427x_{426} + 428x_{427} + 429x_{428} + 430x_{429} + 431x_{430} + 432x_{431} + 433x_{432} + 434x_{433} + 435x_{434} + 436x_{435} + 437x_{436} + 438x_{437} + 439x_{438} + 440x_{439} + 441x_{440} + 442x_{441} + 443x_{442} + 444x_{443} + 445x_{444} + 446x_{445} + 447x_{446} + 448x_{447} + 449x_{448} + 450x_{449} + 451x_{450} + 452x_{451} + 453x_{452} + 454x_{453} + 455x_{454} + 456x_{455} + 457x_{456} + 458x_{457} + 459x_{458} + 460x_{459} + 461x_{460} + 462x_{461} + 463x_{462} + 464x_{463} + 465x_{464} + 466x_{465} + 467x_{466} + 468x_{467} + 469x_{468} + 470x_{469} + 471x_{470} + 472x_{471} + 473x_{472} + 474x_{473} + 475x_{474} + 476x_{475} + 477x_{476} + 478x_{477} + 479x_{478} + 480x_{479} + 481x_{480} + 482x_{481} + 483x_{482} + 484x_{483} + 485x_{484} + 486x_{485} + 487x_{486} + 488x_{487} + 489x_{488} + 490x_{489} + 491x_{490} + 492x_{491} + 493x_{492} + 494x_{493} + 495x_{494} + 496x_{495} + 497x_{496} + 498x_{497} + 499x_{498} + 500x_{499} + 501x_{500} + 502x_{501} + 503x_{502} + 504x_{503} + 505x_{504} + 506x_{505} + 507x_{506} + 508x_{507} + 509x_{508} + 510x_{509} + 511x_{510} + 512x_{511} + 513x_{512} + 514x_{513} + 515x_{514} + 516x_{515} + 517x_{516} + 518x_{517} + 519x_{518} + 520x_{519} + 521x_{520} + 522x_{521} + 523x_{522} + 524x_{523} + 525x_{524} + 526x_{525} + 527x_{526} + 528x_{527} + 529x_{528} + 530x_{529} + 531x_{530} + 532x_{531} + 533x_{532} + 534x_{533} + 535x_{534} + 536x_{535} + 537x_{536} + 538x_{537} + 539x_{538} + 540x_{539} + 541x_{540} + 542x_{541} + 543x_{542} + 544x_{543} + 545x_{544} + 546x_{545} + 547x_{546} + 548x_{547} + 549x_{548} + 550x_{549} + 551x_{550} + 552x_{551} + 553x_{552} + 554x_{553} + 555x_{554} + 556x_{555} + 557x_{556} + 558x_{557} + 559x_{558} + 560x_{559} + 561x_{560} + 562x_{561} + 563x_{562} + 564x_{563} + 565x_{564} + 566x_{565} + 567x_{566} + 568x_{567} + 569x_{568} + 570x_{569} + 571x_{570} + 572x_{571} + 573x_{572} + 574x_{573} + 575x_{574} + 576x_{575} + 577x_{576} + 578x_{577} + 579x_{578} + 580x_{579} + 581x_{580} + 582x_{581} + 583x_{582} + 584x_{583} + 585x_{584} + 586x_{585} + 587x_{586} + 588x_{587} + 589x_{588} + 590x_{589} + 591x_{590} + 592x_{591} + 593x_{592} + 594x_{593} + 595x_{594} + 596x_{595} + 597x_{596} + 598x_{597} + 599x_{598} + 600x_{599} + 601x_{600} + 602x_{601} + 603x_{602} + 604x_{603} + 605x_{604} + 606x_{605} + 607x_{606} + 608x_{607} + 609x_{608} + 610x_{609} + 611x_{610} + 612x_{611} + 613x_{612} + 614x_{613} + 615x_{614} + 616x_{615} + 617x_{616} + 618x_{617} + 619x_{618} + 620x_{619} + 621x_{620} + 622x_{621} + 623x_{622} + 624x_{623} + 625x_{624} + 626x_{625} + 627x_{626} + 628x_{627} + 629x_{628} + 630x_{629} + 631x_{630} + 632x_{631} + 633x_{632} + 634x_{633} + 635x_{634} + 636x_{635} + 637x_{636} + 638x_{637} + 639x_{638} + 640x_{639} + 641x_{640} + 642x_{641} + 643x_{642} + 644x_{643} + 645x_{644} + 646x_{645} + 647x_{646} + 648x_{647} + 649x_{648} + 650x_{649} + 651x_{650} + 652x_{651} + 653x_{652} + 654x_{653} + 655x_{654} + 656x_{655} + 657x_{656} + 658x_{657} + 659x_{658} + 660x_{659} + 661x_{660} + 662x_{661} + 663x_{662} + 664x_{663} + 665x_{664} + 666x_{665} + 667x_{666} + 668x_{667} + 669x_{668} + 670x_{669} + 671x_{670} + 672x_{671} + 673x_{672} + 674x_{673} + 675x_{674} + 676x_{675} + 677x_{676} + 678x_{677} + 679x_{678} + 680x_{679} + 681x_{680} + 682x_{681} + 683x_{682} + 684x_{683} + 685x_{684} + 686x_{685} + 687x_{686} + 688x_{687} + 689x_{688} + 690x_{689} + 691x_{690} + 692x_{691} + 693x_{692} + 694x_{693} + 695x_{694} + 696x_{695} + 697x_{696} + 698x_{697} + 699x_{698} + 700x_{699} + 701x_{700} + 702x_{701} + 703x_{702} + 704x_{703} + 705x_{704} + 706x_{705} + 707x_{706} + 708x_{707} + 709x_{708} + 710x_{709} + 711x_{710} + 712x_{711} + 713x_{712} + 714x_{713} + 715x_{714} + 716x_{715} + 717x_{716} + 718x_{717} + 719x_{718} + 720x_{719} + 721x_{720} + 722x_{721} + 723x_{722} + 724x_{723} + 725x_{724} + 726x_{725} + 727x_{726} + 728x_{727} + 729x_{728} + 730x_{729} + 731x_{730} + 732x_{731} + 733x_{732} + 734x_{733} + 735x_{734} + 736x_{735} + 737x_{736} + 738x_{737} + 739x_{738} + 740x_{739} + 741x_{740} + 742x_{741} + 743x_{742} + 744x_{743} + 745x_{744} + 746x_{745} + 747x_{746} + 748x_{747} + 749x_{748} + 750x_{749} + 751x_{750} + 752x_{751} + 753x_{752} + 754x_{753} + 755x_{754} + 756x_{755} + 757x_{756} + 758x_{757} + 759x_{758} + 760x_{759} + 761x_{760} + 762x_{761} + 763x_{762} + 764x_{763} + 765x_{764} + 766x_{765} + 767x_{766} + 768x_{767} + 769x_{768} + 770x_{769} + 771x_{770} + 772x_{771} + 773x_{772} + 774x_{773} + 775x_{774} + 776x_{775} + 777x_{776} + 778x_{777} + 779x_{778} + 780x_{779} + 781x_{780} + 782x_{781} + 783x_{782} + 784x_{783} + 785x_{784} + 786x_{785} + 787x_{786} + 788x_{787} + 789x_{788} + 790x_{789} + 791x_{790} + 792x_{791} + 793x_{792} + 794x_{793} + 795x_{794} + 796x_{795} + 797x_{796} + 798x_{797} + 799x_{798} + 800x_{799} + 801x_{800} + 802x_{801} + 803x_{802} + 804x_{803} + 805x_{804} + 806x_{805} + 807x_{806} + 808x_{807} + 809x_{808} + 810x_{809} + 811x_{810} + 812x_{811} + 813x_{812} + 814x_{813} + 815x_{814} + 816x_{815} + 817x_{816} + 818x_{817} + 819x_{818} + 820x_{819} + 821x_{820} + 822x_{821} + 823x_{822} + 824x_{823} + 825x_{824} + 826x_{825} + 827x_{826} + 828x_{827} + 829x_{828} + 830x_{829} + 831x_{830} + 832x_{831} + 833x_{832} + 834x_{833} + 835x_{834} + 836x_{835} + 837x_{836} + 838x_{837} + 839x_{838} + 840x_{839} + 841x_{840} + 842x_{841} + 843x_{842} + 844x_{843} + 845x_{844} + 846x_{845} + 847x_{846} + 848x_{847} + 849x_{848} + 850x_{849} + 851x_{850} + 852x_{851} + 853x_{852} + 854x_{853} + 855x_{854} + 856x_{855} + 857x_{856} + 858x_{857} + 859x_{858} + 860x_{859} + 861x_{860} + 862x_{861} + 863x_{862} + 864x_{863} + 865x_{864} + 866x_{865} + 867x_{866} + 868x_{867} + 869x_{868} + 870x_{869} + 871x_{870} + 872x_{871} + 873x_{872} + 874x_{873} + 875x_{874} + 876x_{875} + 877x_{876} + 878x_{877} + 879x_{878} + 880x_{879} + 881x_{880} + 882x_{881} + 883x_{882} + 884x_{883} + 885x_{884} + 886x_{885} + 887x_{886} + 888x_{887} + 889x_{888} + 890x_{889} + 891x_{890} + 892x_{891} + 893x_{892} + 894x_{893} + 895x_{894} + 896x_{895} + 897x_{896} + 898x_{897} + 899x_{898} + 900x_{899} + 901x_{900} + 902x_{901} + 903x_{902} + 904x_{903} + 905x_{904} + 906x_{905} + 907x_{906} + 908x_{907} + 909x_{908} + 910x_{909} + 911x_{910} + 912x_{911} + 913x_{912} + 914x_{913} + 915x_{914} + 916x_{915} + 917x_{916} + 918x_{917} + 919x_{918} + 920x_{919} + 921x_{920} + 922x_{921} + 923x_{922} + 924x_{923} + 925x_{924} + 926x_{925} + 927x_{926} + 928x_{927} + 929x_{928} + 930x_{929} + 931x_{930} + 932x_{931} + 933x_{932} + 934x_{933} + 935x_{934} + 936x_{935} + 937x_{936} + 938x_{937} + 939x_{938} + 940x_{939} + 941x_{940} + 942x_{941} + 943x_{942} + 944x_{943} + 945x_{944} + 946x_{945} + 947x_{946} + 948x_{947} + 949x_{948} + 950x_{949} + 951x_{950} + 952x_{951} + 953x_{952} + 954x_{953} + 955x_{954} + 956x_{955} + 957x_{956} + 958x_{957} + 959x_{958} + 960x_{959} + 961x_{960} + 962x_{961} + 963x_{962} + 964x_{963} + 965x_{964} + 966x_{965} + 967x_{966} + 968x_{967} + 969x_{968} + 970x_{969} + 971x_{970} + 972x_{971} + 973x_{972} + 974x_{973} + 975x_{974} + 976x_{975} + 977x_{976} + 978x_{977} + 979x_{978} + 980x_{979} + 981x_{980} + 982x_{981} + 983x_{982} + 984x_{983} + 985x_{984} + 986x_{985} + 987x_{986} + 988x_{987} + 989x_{988} + 990x_{989} + 991x_{990} + 992x_{991} + 993x_{992} + 994x_{993} + 995x_{994} + 996x_{995} + 997x_{996} + 998x_{997} + 999x_{998} + 1000x_{999} + 1001x_{1000} + 1002x_{1001} + 1003x_{1002} + 1004x_{1003} + 1005x_{1004} + 1006x_{1005} + 1007x_{1006} + 1008x_{1007} + 1009x_{1008} + 1010x_{1009} + 1011x_{1010} + 1012x_{1011} + 1013x_{1012} + 1014x_{1013} + 1015x_{1014} + 1016x_{1015} + 1017x_{1016} + 1018x_{1017} + 1019x_{1018} + 1020x_{1019} + 1021x_{1020} + 1022x_{1021} + 1023x_{1022} + 1024x_{1023} + 1025x_{1024} + 1026x_{1025} + 1027x_{1026} + 1028x_{1027} + 1029x_{1028} + 1030x_{1029} + 1031x_{1030} + 1032x_{1031} + 1033x_{1032} + 1034x_{1033} + 1035x_{1034} + 1036x_{1035} + 1037x_{1036} + 1038x_{1037} + 1039x_{1038} + 1040x_{1039} + 1041x_{1040} + 1042x_{1041} + 1043x_{1042} + 1044x_{1043} + 1045x_{1044} + 1046x_{1045} + 1047x_{1046} + 1048x_{1047} + 1049x_{1048} + 1050x_{1049} + 1051x_{1050} + 1052x_{1051} + 1053x_{1052} + 1054x_{1053} + 1055x_{1054} + 1056x_{1055} + 1057x_{1056} + 1058x_{1057} + 1059x_{1058} + 1060x_{1059} + 1061x_{1060} + 1062x_{1061} + 1063x_{1062} + 1064x_{1063} + 1065x_{1064} + 1066x_{1065} + 1067x_{1066} + 1068x_{1067} + 1069x_{1068} + 1070x_{1069} + 1071x_{1070} + 1072x_{1071} + 1073x_{1072} + 1074x_{1073$$

$\rightarrow \max$
Розв'яжемо цю задачу симплекс-методом.
Складемо симплекс-таблицю:

			B	7	3	0	0
1		0	20	5	2	1	0
2		0	19	4	2	0	1
			0	-7	-3	0	0
1		7	4	1			0
2		0	3	0			1
			28	0			0
1		7	1	1	0	1	-1
2		3		0	1	-2	
				0	0	1	

Умова ; оптимальності ; виконується, бо всі
. В оптимальному плані задачі є дробова

координата . Змінна , яка приймає не цілочислове значення знаходиться в другому рядку останньої симплекс-таблиці. З допомогою коефіцієнтів цього рядка й будується відсікання Гоморі.

,

Це рівняння додається в останню таблицю для продовження розв'язування.

			B	7	3	0	0	0
1		7	1	1	0	1	-1	0
2		3		0	1	-2		0
3		0	-1	0	0	0	-1	1
				0	0	1		0
1		7	2	1	0	1	0	-1
2		3	5	0	1	-2	0	
3		0	1	0	0	0	1	-1
			29	0	0	1	0	

Розв'язувальний елемент у розширеній симплекс-таблиці ми вибрали, використовуючи процедуру двоїстого симплексного методу. З останньої симплекс-таблиці ми одержали оптимальний цілочисловий розв'язок даної задачі.

, тобто ; .

Відповідь:

Для частковоцілочислової задачі додаткові відсікання будуються лише для тих змінних, на які накладена умова цілочисловості. Якщо для деякої змінної, що на неї накладено вимогу цілочисловості, отримано дробовий результат b_s , тоді організуємо відтинання щодо s -того рядка. Дане відсікання

запишемо у вигляді додаткового рядка з балансуючою змінною $x_{n+1} \geq 0$:

де $a_j, j=1, \dots, n$, - спеціально підраховані коефіцієнти. Для змінних x_j , на які не накладено вимог цілочисловості:

а для змінних x_j з вимогою цілочисловості (5)

Додатково введений рядок беремо за розв'язувальний, і після вибору провідного елемента за принципом двоїстого симплекс-методу переходимо до нової симплекс-таблиці. (6)

10 Транспортна задача

10.1 Постановка транспортної задачі

Загальна постановка транспортної задачі полягає у знаходженні оптимального плану перевезень деякого однорідного вантажу з m пунктів відправлення у n пунктів призначення. При цьому критерієм оптимальності є або мінімальна вартість перевезень усього вантажу, або мінімальний час його доставки. Ми розглядатимемо випадок, де критерієм оптимальності є мінімальна вартість перевезень.

Позначимо через x_{ij} кількість одиниць вантажу, який перевозиться з пункту відправлення i в пункт призначення j або від i -того постачальника до j -того споживача, через a_i —запаси вантажу в пункті i , через b_j — потреби у вантажі в пункті j , через c_{ij} — вартість

перевезення одиниці вантажу з пункту в пункт .
Тоді математична модель транспортної задачі має вигляд:

Якщо в транспортній задачі загальна кількість продукції постачальників дорівнює загальному попиту всіх споживачів,

тобто
то таку транспортну задачу ; називають **закритою** або **збалансованою**.

Якщо , то транспортну задачу називають **відкритою** або **незбалансованою**.

Будь-який допустимий (невід'ємний) розв'язок системи обмежень , транспортної задачі називають **планом транспортної задачі** і позначають матрицею

. Оптимальним планом транспортної задачі називають матрицю , яка

задовольняє умови задачі і для якої цільова функція досягає найменшого значення.

Теорема 1. (Умова існування розв'язку транспортної задачі): необхідною і достатньою умовою існування розв'язку транспортної задачі є її закритість (збалансованість), тобто

10.2 Метод потенціалів знаходження розв'язку транспортної задачі

Транспортна задача є задачею лінійного програмування, яку можна розв'язати симплекс-методом. Але специфічна структура транспортної задачі дає змогу використовувати для її розв'язання більш ефективний метод, який повторює, по суті, кроки симплекс-алгоритму. Таким є метод потенціалів.

Алгоритм методу розв'язання транспортної задачі складається з таких етапів:

1. Визначення типу транспортної задачі (відкрита чи замкнута).
2. Побудова першого опорного плану транспортної задачі.
3. Перевірка плану транспортної задачі на оптимальність.
4. Якщо умова оптимальності виконується, то маємо оптимальний розв'язок транспортної задачі. Якщо ж умова оптимальності не виконується, необхідно перейти до наступного опорного плану.
5. Новий план знову перевіряємо на оптимальність, тобто повторюємо дії п.3 і т.д.

Розглянемо кожний етап цього алгоритму.

Якщо при перевірці збалансованості виявилось, що

, то задачу треба звести до замкнутого виду, тобто зрівняти поставки і потреби. Робиться це введенням фіктивного постачальника або фіктивного споживача залежно

від співвідношення сум $\sum_{j=1}^n a_j$ і $\sum_{i=1}^m b_i$.
Можливі два випадки:

а) Якщо $\sum_{j=1}^n a_{ij} < b_i$, то водиться під номером i -ий

фіктивний постачальник із поставкою $b_i - \sum_{j=1}^n a_{ij}$.

б) Якщо $\sum_{i=1}^m a_{ij} < b_j$, то вводиться j -ий фіктивний

споживач з потребою $b_j - \sum_{i=1}^m a_{ij}$. Вартості перевезення одиниці продукції для фіктивного постачальника або фіктивного споживача вважаються рівними нулю.

Такий вибір цих вартостей при поставках не впливає на значення цільової функції.

1. Забезпечивши закритість розв'язуваної задачі, розпочинаємо будувати початковий опорний план. Із математичної моделі транспортної задачі очевидно, що число змінних дорівнює $m+n$, а число рівнянь $m+n-1$. Оскільки ми припускаємо, що виконується умова $m+n-1 \leq m+n$, то число лінійно незалежних рівнянь дорівнює $m+n-1$. Тому опорний план може мати не більше, ніж $m+n-1$ додатних координат. Якщо цих координат точно $m+n-1$, то план називається **невиродженим**, якщо ж менше – **виродженим**. В такому випадку, якщо план вироджений і число заповнених клітинок таблиці транспортної задачі менше $m+n-1$, то необхідно доповнити таблицю до кількості $m+n-1$ заповнених клітинок, ввівши в порожні клітинки об'єм перевезень, рівний нулеві.

Зручно подавати транспортну задачу і її розв'язування за допомогою транспортної таблиці (матриці перевезень), яка має такий вигляд:

			..		запаси
			..		

...	..	.	..	.	..	...
			..			
...	.	.	.	.	.	...
			..			
потреб и			..			

Для побудови опорних планів задачі (1)-(4) використовують такі три методи: північно-західного кута, мінімальної вартості, подвійної переваги.

1. Метод північно-західного кута.

Згідно з цим методом побудову опорного плану починають з заповнення лівої верхньої клітинки таблиці, спочатку розподіляють запаси першого постачальника, причому він має спочатку задовольнити якомога повніше потреби першого споживача. Якщо запаси залишаються, то він має задовольнити потреби другого споживача і т.д. закінчують заповнення таблиці в правій нижній клітинці.

2. Метод мінімальної вартості.

Ідея цього методу полягає в тому, що на кожному кроці заповнюють клітинку таблиці, яка має найменшу вартість перевезень одиниці вантажу. Такі дії повторюють до тих пір, поки не буде розподілено всю продукцію між постачальниками і споживачами.

При користуванні цим методом з розгляду виключаємо рядок, який відповідає постачальнику, запаси якого повністю використано або стовпчик, що відповідає споживачу, потреби якого повністю задоволено.

3. Метод подвійної переваги.

Цей метод застосовується в транспортних задачах з великою кількістю перевезень. Спочатку в кожному рядку позначаємо клітинку з найменшою вартістю знаком , потім

таке саме робимо для кожного стовпчика таблиці. Отримуємо окремі клітинки, що мають дві позначки, деякі – по одній, а деякі зовсім не мають позначок. Розподіл вантажу починаємо з клітинки з двома позначками, вибравши з них клітинку з мінімальною вартістю і рухаємося в міру зростання вартості перевезень. Далі заповнюємо клітинки позначені один раз, а вже потім – непозначені клітинки, користуючись при цьому методом мінімальної вартості.

1 Опорний план перевіряють на оптимальність за

допомогою потенціалів і відповідно постачальників і споживачів.

Теорема 2. (умова оптимальності опорного плану транспортної задачі): якщо для деякого опорного плану існують числа і, для яких виконуються умови:

$$i=1, \dots, m; \text{ та } j=1, \dots, n,$$

то він є оптимальним планом транспортної задачі.

Числа і називаються потенціалами відповідно пунктів відправлення і пунктів призначення.

Алгоритм методу потенціалів

1. Числа і знаходимо із системи рівнянь

де – вартості, які стоять у заповнених клітинках таблиці умови задачі. Таких рівнянь є, а невідомих, тобто число рівнянь на одиницю менше числа невідомих, а тому одне з невідомих, наприклад, можна взяти рівним нулю. Усі інші і знаходимо з системи.

2. Після цього для вільних клітинок знаходимо числа

Якщо серед цих чисел немає додатних, то одержимо оптимальний план. Випишемо одержаний розв'язок і обчислимо вартість перевезень.

3. Якщо ж є додатні числа серед обчислених, то переходимо до нового опорного плану. Для цього серед чисел вибирають найбільше і для клітинки, якій воно відповідає, роблять перерахунок вантажу за циклом, тобто заповнюють цю клітинку, зробивши перерозподіл вантажу між клітинками, які пов'язані з нею циклом.

Цикл перерахунку вільної клітинки є замкнута ламана лінія з горизонтальними і вертикальними ланками, що чергуються; одна із вершин ламаної знаходиться у вільній клітинці, а всі інші вершини ламаної знаходяться у завантажених клітинках.

4. Послідовно розставляємо „+” і „-“ у вершинах циклу, починаючи від клітинки, для якої здійснюється перерозподіл.

Серед об'ємів поставок в клітинках із знаком „-“ вибираємо найменше значення і додаємо його до об'ємів поставок в клітинках з знаком „+” та віднімаємо від об'ємів поставок в клітинках зі знаком „-“.

5. Будуємо новий опорний план, який досліджуємо на оптимальність згідно вказаного алгоритму.

Приклад 10.1.

Знайти опорний план даної транспортної задачі методом північно-західного кута.

				запас и
	4	1	5	18
	2	3	6	10
	5	7	4	20
потреб и	25	10	13	48

, отже, дана транспортна задача закрыта.

В клітинку помістимо
Вантаж першого постачальника вичерпаний, тому перший рядок виключається з розрахунку. Переходимо до розподілу вантажу другого постачальника. У клітинки помістимо необхідну кількість вантажу. Попит першого споживача задоволено повністю, перший стовпчик виключається з розрахунку. Залишок вантажу другого постачальника записуємо в клітинку :
Вантаж другого постачальника вичерпаний, другий рядок виключається з розрахунку. Переходимо до розподілу вантажу третього постачальника. З урахуванням потреб другого споживача в клітинку поміщаємо необхідну кількість вантажу. Потреби другого споживача задоволено повністю. У клітинку поміщаємо залишок вантажу третього постачальника. Одержали повний розподіл запасів вантажу.

				запаси
	4 18	1 –	5 –	18
	2 7	3 3	6 –	10
	5 –	7 7	4 13	20
потреби	25	10	13	48

Заповненими виявилися 5 клітинок, що дорівнює , а це означає, що даний план невироджений.

Приклад 10.2.

Побудувати за правилом подвійної переваги початковий опорний план транспортної задачі.

					запаси
	5	4	1	2	60
	4	2	6	3	40
	7	3	5	4	35
потреби	40	25	20	50	135

В першому рядку ми відзначимо клітинку (1;3) значком , у якій стоїть найменша вартість. В другому рядку відмічаємо кліти (2;2), у третьому – (3;2). Аналогічно зробимо і в стовпчиках. Таким чином, в клітинках (1;3) і (2;2) буде 2 позначки, тому їх заповнювати будемо першими. Потім заповнюємо клітинки (1;4) і (2;1) з однією позначкою. Далі заповнюємо всі інші клітинки і одержимо таку таблицю:

					запаси
	5 –	4 –	1 20	2 40	60
	4 15	2 25	6 –	3 –	40
	7 25	3 –	5 –	4 –	35
потреби	40	25	20	50	135

Даний план – не вироджений. Число заповнених клітинок буде 6, а це дорівнює .

Приклад 10.3.

У пунктах постачання , , є однорідний вантаж в обсязі 45, 75, 80 одиниць відповідно; цей вантаж треба транспортувати у пункти , , в обсязі відповідно 35, 85, 80 одиниць. Матриця перевезень одиниці вантажу має вигляд

Знайти оптимальний план перевезень, тобто такий план для якого сума транспортних витрат є мінімальною. Позначимо через – кількість одиниць вантажу, який перевозиться з пункту в пункт . Матриця перевезень транспортної задачі має вигляд:

				запаси
	2	5	7	45
	4	18	24	75
	6	30	22	80
потреби	35	85	80	200

, отже, дана транспортна задача замкнута.

Математична модель цієї задачі запишеться так:

Загальна вартість перевезень є

Потрібно серед всіх допустимих розв'язків записаної системи обмежень знайти такий, при якому цільова функція досягає мінімального значення.

Транспортна задача розв'язується симплекс-методом з використанням матриці перевезень.

Спочатку знаходимо опорний план транспортної задачі, наприклад, методом найменшої вартості.

В нашому випадку дістанемо:

Потенц.	2	5	11	запаси
0	2 35	5 10	7 -	45
-13	4 -	18 75	24 0	75
-11	6 -	30 -	22 80	80
потреби	35	85	80	200

Заповнені (базисні) клітинки відповідають базисним невідомим, а незаповнені (вільні) клітинки – вільним невідомим.

В нас заповнених клітинок 4, а . Отже, опорний план вироджений. У такому разі необхідно заповнити

вільну клітинку, помістивши туди нульовий вантаж. Нехай це буде клітинка . Опорний план став невиродженим. Перевіримо цей план на оптимальність. Потенціали постачальників і споживачів знайдемо, розв'язавши таку систему рівнянь

, записаних для всіх заповнених клітинок:

Покладемо .
Розв'язавши цю систему рівнянь, одержимо:

$$\begin{aligned} & ; \\ & ; \quad ; \\ & ; \quad . \end{aligned}$$

Для вільних клітинок знаходимо числа :

$$\begin{aligned} & ; \\ & ; \\ & ; \\ & . \end{aligned}$$

План неоптимальний, бо є .

Серед додатних вибираємо найбільше . Для клітинки (2;1), якій воно відповідає, робимо перерахунок вантажу за циклом і переходимо до нового опорного плану: починаючи з клітинки (2;1), проставляємо чергуючи знаки „+”

і „–“. Серед вершин циклу із знаком „–“ вибираємо ту , в якій об’єм поставок найменший, тобто , додаємо його до вантажу в клітинках із знаком „+” і віднімаємо від вантажу в клітинках із знаком „–“. Одержуємо новий опорний план, який є невід’одженим.

Потенц.	2	5	11	запаси
0	2 –	5 45	7 –	45
-13	4 35	18 40	24 0	75
-11	6 –	30 –	22 80	80
потреби	35	85	80	200

Перевіряємо його на оптимальність.
Знайдемо потенціали постачальників і споживачів, записавши відповідну систему для заповнення клітинок:

Якщо , то ;
;
;

Для вільних клітинок знаходимо числа .

;
;

План неоптимальний, бо є .

Для клітинки (1;3) будемо цикл і аналогічно до попередньому випадку робимо перерахунок вантажу за циклом. Одержимо новий невикористаний опорний план.

Потенц.	-9	5	7	запаси
0	2 —	5 45	7 0	45
-13	4 35	18 40	24 —	75
-15	6 —	30 —	22 80	80
потреби	35	85	80	200

Перевіримо цей план на оптимальність.

Запишемо систему рівнянь для заповнених клітинок, розв'яжемо її і знайдемо потенціали:

Якщо взяти $u_1 = 0$, то $u_2 = 9$;
 $u_3 = 5$;
 $u_4 = 7$;
 $u_5 = 45$.

Знайдемо v_j для вільних клітинок:

$v_1 = 0$;
 $v_2 = 0$;
 $v_3 = 0$;

Таким чином, одержаний план є оптимальним.

Він має вигляд
і йому відповідає мінімальне значення цільової функції:

Приклад 10.4

Для будівництва чотирьох об'єктів використовується цегла, яка виготовляється на трьох заводах. Щоденно кожний завод може виготовити 100, 150 і 50 m^3 цегли. Щоденні потреби в цеглі на кожному з об'єктів відповідно дорівнюють 75, 80, 60 і 85 m^3 .

Відомі також тарифи перевезень:

Скласти такий план перевезень цегли до об'єктів, при якому загальна вартість перевезень буде мінімальною.

Матриця перевезень даної задачі має вид:

					запаси
	6	7	3	5	100
	1	2	5	6	150
	8	10	20	1	50
потреби	75	80	60	85	300

, отже, транспортна задача є закритою.

Нехай x_{ij} – кількість m^3 цегли, яка перевозиться з i -го заводу до j -го об'єкту.

Математична модель задачі запишеться так:

Загальна вартість перевезень:

Серед усіх допустимих розв'язків системи обмежень знайти такий, при якому цільова функція мінімізується.

Знайдемо вихідний опорний план цієї задачі методом найменшої вартості.

Потенціал и	6	7	3	5	запаси
0	6 —	7 5	3 60	5 35	100
5	1 75	2 75	5 —	6 —	150
4	8 —	10 —	20 —	1 50	50
потреби	75	80	60	85	300

Заповнених базисних клітинок є шість. Опорний план невикористаний. Перевіримо його на оптимальність. Складемо

систему для знаходження потенціалів:

Розв'язавши дану систему при умові, що ,
 одержимо ; ; ; ; .
 Для кожної вільної клітинки обчислимо числа
 :

,
 ,
 ,
 ,
 ,

Серед цих чисел немає додатних, отже, одержаний план
 оптимальний.

Відповідь: ,

Приклад 10.5.

Знайти оптимальний розв'язок транспортної задачі

				запаси
	0	0	8	40

	3	5	1	60
	1	2	4	50
потреби	70	30	20	150
				120

Дана транспортна задача – це задача з неправильним балансом. Так як , то вводимо фіктивного споживача із потребою . Якщо додано фіктивного споживача, то в матриці перевезень виникає додатковий стовпець, вартості перевезень у цих фіктивних клітинках беремо нульовими. Поставлена задача стає закритою.

Матриця перевезень має вид:

					запаси
	0	0	8	0	40
	3	5	1	0	60
	1	2	4	0	50
потреби	70	30	20	30	150

Математична модель задачі:

Загальна вартість перевезень буде:

Методом найменшої вартості знайдемо опорний план:

Потенц.	0	1	-3	-4	запаси
0	0 40	0 —	8 —	0 —	40
-4	3 —	5 10	1 20	0 30	60
-1	1 30	2 20	4 —	0 —	50
потреби	70	30	20	30	150

Знайдений опорний план не вироджений. Перевіримо його на оптимальність. Для цього знайдемо потенціали, склавши систему рівнянь для заповнених клітинок:

Якщо покладемо $d_{12}=1$ і $d_{21}=1$, то система матиме такий розв'язок:

Для вільних клітинок знайдемо числа.

;
,
,
,
,
,

Одержаний план неоптимальний, бо серед чисел d_{ij} є додатні, а саме $d_{12}=1$ і $d_{21}=1$. Завантажимо одну із цих клітинок, а саме (1;2). Побудуємо для неї цикл і зробимо перерахунок вантажу за циклом, тобто заповнюємо цю клітинку, зробивши перерозподіл вантажу між клітинками, які пов'язані з нею циклом. Отримуємо новий опорний план.

Потенц.	0	1	-4	-5	запаси
0	0 20	0 20	8 —	0 —	40
-4	3 —	5 10	1 20	0 30	60
-1	1	2	4	0	50

	50	–	–	–	
потреби	70	30	20	30	150

Запишемо систему _____ для базисних клітинок:

Покладемо _____.

Розв'язавши систему, одержимо: _____;
 _____;
 _____;
 _____;

Знайдемо для вільних клітинок числа _____:

_____,
 _____,
 _____,
 _____,
 _____,
 _____.

Серед чисел _____ є додатне _____. З цієї клітинки
 будемо цикл і робимо перерахунок вантажу по циклу і
 переходимо до нового опорного плану.

Потенц.	0	1	-4	-5	запаси
---------	---	---	----	----	--------

0	0 10	0 30	8 –	0 –	40
-4	3 10	5 –	1 20	0 30	60
-1	1 50	2 –	4 –	0 –	50
потреби	70	30	20	30	150

Одержаний план не вироджений. Перевіримо цей план на оптимальність. Система для знаходження потенціалів

запишеться так:

Якщо покладемо $x_1 = 1$, то розв'язок системи буде такий:

$$x_2 = 1; x_3 = 1; x_4 = 1; x_5 = 1; x_6 = 1.$$

Для вільних клітинок обчислимо числа x_7 :

$$x_7 = 1, \\ x_8 = 1, \\ x_9 = 1, \\ x_{10} = 1, \\ x_{11} = 1, \\ x_{12} = 1.$$

Опорний план оптимальний, бо всі $x_i \geq 0$.

11 Матричні ігри

11.1 Вступ в теорію ігор

Теорія ігор займається вивченням конфліктних ситуацій та розробкою математичних методів вирішення цих ситуацій. Конфліктні ситуації часто виникають в умовах ринкової економіки, тому що колективи (або окремі фізичні особи) часто мають антагоністичні (протилежні) цілі та інтереси.

Характерною особливістю ігрової ситуації є взаємодія інтересів 2-х чи більше сторін, які діють як суперники. Кожна

сторона для досягнення цілей старається оптимізувати свої рішення.

Основні ідеї теорій ігор появилися в 1-ій половині 20 століття в працях математиків Е.Бореля, Дж. фон Неймана та економіста О.Моргенштерна.

Кожна гра характеризується наявністю **правил** (дій гравців, спрямованих на досягнення своєї мети). Учасники гри обирають певну **стратегію** (лінію поведінки). Гравці мають на меті отримати в результаті певний **виграш**.

Будемо розглядати ігри з двома гравцями, в якій сумарний виграш одного гравця дорівнює сумарному програшу іншого. Таку гру, коли арифметична сума виграшів (програш вважається виграшем із знаком “-”) дорівнює нулю, називають **грою з нульовою сумою**. Ситуації, що моделюються такими іграми, є типовими для практичної діяльності фірм, менеджерів, маркетологів та інших спеціалістів, що діють в умовах ринку.

Нехай маємо гру двох осіб I і II з нульовою сумою.

Назвемо S_I – можливі стратегії гравця I ; S_{II} – можливі стратегії гравця II ; P_I ; P_{II} . Інтереси кожного гравця можна врахувати з допомогою цільової функції, яка називається *функцією виграшу* (або *функцією платежу*).

Нехай $U_I(s_I, s_{II})$ – функція виграшу гравця I ;

$U_{II}(s_I, s_{II})$ – функція виграшу гравця II ; $U_I(s_I, s_{II}) + U_{II}(s_I, s_{II}) = 0$.

За умовою, $U_I(s_I, s_{II}) = -U_{II}(s_I, s_{II})$ (гра з нульовою сумою).

Тоді очевидно, що $U_I(s_I, s_{II}) = -U_{II}(s_I, s_{II})$.

Таким чином, мета гравця I – максимізувати функцію $U_I(s_I, s_{II})$, а гравця II – мінімізувати її.

Нехай $U = (U_I, U_{II})$. Тоді отримаємо матрицю:

-ті рядки матриці відповідають стратегіям гравця
; а -ті стовбці – стратегіям гравця .

Матриця називається *платіжною* (також матрицею гри). Елементом a_{ij} цієї матриці є виграш гравця , якщо він вибрав стратегію A_i , а гравець – стратегію B_j . Відповідно цей же елемент a_{ij} є програвом гравця . Якщо $a_{ij} < 0$, то гравець має від'ємний виграш (тобто програш), а гравець – відповідно від'ємний програш (виграш).

11.2 Чисті стратегії та цілком виражені ігри

Із багатьох критеріїв, які пропонуються теорією ігор для вибору оптимальних стратегій, найпоширенішим є *критерій мінімаксу-максиміну*.

Сутність його полягає в наступному.

Нехай гравець вибирає стратегію . Тоді в найгіршому для себе випадку він отримає виграш, що дорівнює . Якщо навіть гравець знає його стратегію, гравець повинен діяти так, щоб максимілізувати свій мінімальний виграш.

Позначимо цей виграш . Тоді .

Таку стратегію гравця називають *максимінною*. Розмір його гарантованого мінімального виграшу називають **нижньою ціною гри**.

Гравець , який програє суму в розмірі елементів , повинен обрати стратегію, що мінімізує його максимально можливий програш (як би не діяв гравець). Стратегію гравця називають *мінімаксною*. Розмір його програшу позначається (програш може бути і від'ємний, тоді він

отримає виграш). Число називається **верхньою ціною гри**, і аналогічні міркування щодо стратегії гравця дають:

Оптимальний розв'язок цієї задачі для кожного гравця досягається тоді, коли кожному з них не вигідно змінювати обрану стратегію, оскільки суперник може вибрати у відповідь іншу стратегію, яка дасть йому кращий результат.

Якщо , то гра називається **цілком визначеною**. Цілком визначені ігри ще називаються **іграми з сідловою точкою** (або сідловими точками). У такій грі оптимальним для обох гравців є вибір чистих стратегій – максимінної для гравця і мінімаксної для гравця .

11.3 Змішані стратегії

Якщо гра не має сідлової точки, тобто і , то максимінно-мінімаксні стратегії не є оптимальними і кожний гравець може покращити свій результат, обираючи якимось чином інші стратегії. Тоді оптимальний розв'язок такої гри знаходять, застосовуючи *змішані стратегії*, які є певними комбінаціями чистих стратегій.

Вектор, компоненти якого показують ймовірності доцільного використання гравцем своїх чистих стратегій у багаторазовій грі, називають *змішаною стратегією*. Тоді завдання змішаних стратегій полягає у записуванні тих ймовірностей, з якими вибираються його початкові стратегії: для гравця та для гравця . З самого означення частот (або

ймовірностей) випливає, що . Розглянемо приклад гри, яка має сідлову точку.

Приклад 11.1

Нехай гра двох гравців і задана платіжною матрицею:

Визначити ціну гри та оптимальні стратегії для гравців.
З умови маємо:

Матрицю можна попередньо спростити. Назвемо чисту стратегію A_s гравця **домінуючою** над чистою стратегією A_k , якщо всі елементи рядка, що відповідають стратегії A_s , не менші за відповідні елементи рядка, що відповідають стратегії A_k . **Домінуючою** стратегією B_r над стратегією B_l для гравця B називається та, при якій елементи r -го стовпця не більші, ніж відповідні елементи l -го стовпця. В нашій матриці видно, що елементи четвертого рядка не менші за відповідні елементи третього рядка. Тоді третій рядок матриці можна не враховувати (викреслити).

Тоді будемо мати:

Таким чином .

Отже, ціна гри . Сідловим елементом є , що знаходиться на перетині четвертого рядка та четвертого стовпця початкової матриці. Отже, оптимальними чистими стратегіями для гравців є та .

Відповідь: .

11.4 Зведення матричної гри для двох гравців до задачі лінійного програмування

Нехай матрична гра не має сідлової точки. Тоді $\alpha < \beta$. Тоді знаходження розв'язку, особливо платіжної матриці великих розмірів (тобто за великої кількості стратегії для гравців), є досить складною задачею. Таку задачу можна успішно розв'язати методом лінійного програмування, для чого потрібно сформулювати відповідну математичну модель такої задачі.

Сформулюємо задачу таким чином: знайти вектори ймовірностей (або те саме, що вектори частот)

та , які і вказують ймовірності вибору стратегій для кожного гравця .

Без доведення зазначимо *основну теорему теорії ігор*: кожна скінченна гра має розв'язок, який можливий в області змішаних стратегій.

Нехай дано скінченну матричну гру з платіжною матрицею:

Оптимальні стратегії гравців дозволяють отримати виграш u^0 , тоді використання оптимальної змішаної стратегії гравцем i має забезпечувати виграш, не менший за ціну гри (при виборі гравцем j будь-яких стратегій).
Математично така умова має вигляд:

$$u^0 \leq \sum_{j=1}^n x_j a_{ij} \quad ; \quad j=1, \dots, n$$
Відповідно використання оптимальної змішаної стратегії гравцем j має забезпечувати йому програш, що не перевищує (при будь-якому виборі стратегій гравцем i):

$$z^0 \geq \sum_{i=1}^m y_i a_{ij} \quad ; \quad i=1, \dots, m$$
Співвідношення (2) і (3) використовують для знаходження розв'язку гри.

Якщо u^0 і z^0 – оптимальні стратегії гравців, то число u^0 є ціною гри.

Можна вважати, що u^0 є ціною гри. Якщо це не так, то додавши до всіх елементів матриці A константу c , одержимо A' . Це не приведе до зміни оптимальних стратегій, а лише збільшить ціну гри на c .

Поділимо обидві частини A і A' на c і введемо позначення:

$$a_{ij} = \frac{a_{ij}}{c} \quad ; \quad i=1, \dots, m \quad ; \quad j=1, \dots, n$$
Тоді дістанемо співвідношення:

$$\begin{aligned} u^0 &\leq \sum_{j=1}^n x_j a_{ij} \quad ; \quad j=1, \dots, n \\ z^0 &\geq \sum_{i=1}^m y_i a_{ij} \quad ; \quad i=1, \dots, m \end{aligned}$$

Очевидно, що
Тоді, врахувавши (4), одержимо:

Гравець i намагається дістати максимальний виграш, а гравець j – мінімальний програш. З цього випливає:

Враховуючи (5), (6), (7), одержимо такі задачі лінійного програмування:

;

(8)

і двоїста: $j=1, \dots, n$

;

(9)

, $i=1, \dots, m$

Якщо знайдені оптимальні розв'язки x_i і y_j задач (8) і (9), то змішані стратегії для гравців і ціна гри знаходиться за

формулами:

$$x_i = \frac{a_i}{\sum_{j=1}^n a_j} ; \quad y_j = \frac{b_j}{\sum_{i=1}^m b_i} ; \quad i=1, \dots, m ; j=1, \dots, n$$

;

x_i – відповідно координати векторів x та y , що є оптимальними розв'язками задач (8) і (9).

Приклад 11.2

Розв'язати матричну гру, задану платіжною матрицею:

Розв'язання:

Спочатку перевіримо, чи має матрична гра розв'язок в чистих стратегіях:

Отже, гра не має розв'язку в чистих стратегіях,

За матрицею гри запишемо пару двоїстих задач лінійного програмування:

;

; $j=1,2,3$.
;

, $i=1,2,3$.
Розв'яжемо вихідну задачу симплексним методом:
;

; $j=1,\dots,6$.
Складемо симплекс-таблицю до задачі :

i		$C_{\bar{b}}$		1	1	1	0	0	0
1		0	1	1	2	0	1	0	0
2		0	1	1	0	1	0	1	0
3		0	1	2	1	0	0	0	1
$m+1$			0	-1	-1	-1	0	0	0
1		0	1	1	2	0	1	0	0
2		1	1	1	0	1	0	1	0
3		0	1	2	1	0	0	0	1
$m+1$			1	0	-1	0	0	1	0
1		1			1	0		0	0
2		1	1	1	0	1	0	1	0

3		0			0	0		0	1
$m+1$					0	0		1	0

Маємо оптимальний розв'язок задачі :

;

Тоді ;

.

Перевірка: ;

Відповідь: ціна гри ; вектори оптимальних змішаних

стратегій – для гравця : ; для гравця :

12 Графічний метод розв'язування задач нелінійного програмування

Лінійні математичні моделі можуть виявитися недостатніми для реальних економічних процесів. Наприклад, такі показники, як прибуток, собівартість, капітальні витрати на виробництво та інші нелінійно залежать від обсягу виробництва, витрат ресурсів та робочого часу. Це приводить до необхідності побудови нелінійних економіко-математичних моделей.

Нехай задана функція

Загальною задачею математичного програмування називається задача, в якій потрібно знайти найбільше (або найменше) значення функції $f(x)$, тобто

у припущенні, що на змінні x_i накладається обмеження

$x_i \geq 0$; $i=1, \dots, m$.
Крім того, в задачах математичного програмування звичайно накладаються обмеження на знаки змінних:

$x_j \leq 0$; $j=1, \dots, n$.
Таким чином, маємо задачу (1)–(3).

Система обмежень (2) називається **областю допустимих розв'язків, або множиною обмежень**. Точка називається **допустимим розв'язком** задачі – .

Допустимий розв'язок x^* , при якому функція $f(x)$ досягає максимуму (мінімуму), називають оптимальним розв'язком.

Будемо розглядати задачі, в яких хоча б одна з функцій $f(x)$ або $g_i(x)$ з умов $g_i(x) \leq 0$, є нелінійною. Це, власне, і буде **задачею нелінійного програмування**.

За своїми властивостями задачі нелінійного програмування суттєво відрізняються від задач лінійного програмування. Наприклад, область допустимих планів може не бути опуклою, а екстремум цільової функції може досягатися в будь-якій точці області допустимих розв'язків.

Для задач нелінійного програмування не існує універсального методу розв'язування (як, наприклад, симплексний метод для задач лінійного програмування). Тому для задач нелінійного програмування щоразу треба використовувати притаманний конкретній задачі метод.

Графічний метод є особливо зручним для випадку двох

змінних, тобто x_1, x_2 . Тоді множину обмежень можна зобразити як множину точок на площині.

Розв'яжемо задачу нелінійного програмування графічним методом.

Приклад 12.1

Знайти максимальне значення функції

якщо



Розв'язання.

Побудуємо область допустимих розв'язків, маємо чотирикутник (рис. 12.1):

Перетворимо :



Лінії рівня функції

тобто

або

Отже, лінії рівня – параболи, що мають вісь симетрії

3 8 C

Рисунок 12.1

Максимального значення цільова функція буде досягати в точці дотику $E(3;4)$ виділеної параболи до області допустимих розв'язків.

Тоді маємо:

Відповідь:

Приклад 12.2

Знайти максимальне і мінімальне значення функції

якщо

Розв'язання

Накреслимо область допустимих розв'язків, маємо п'ятикутник $ABCD$, (рис.12.2):

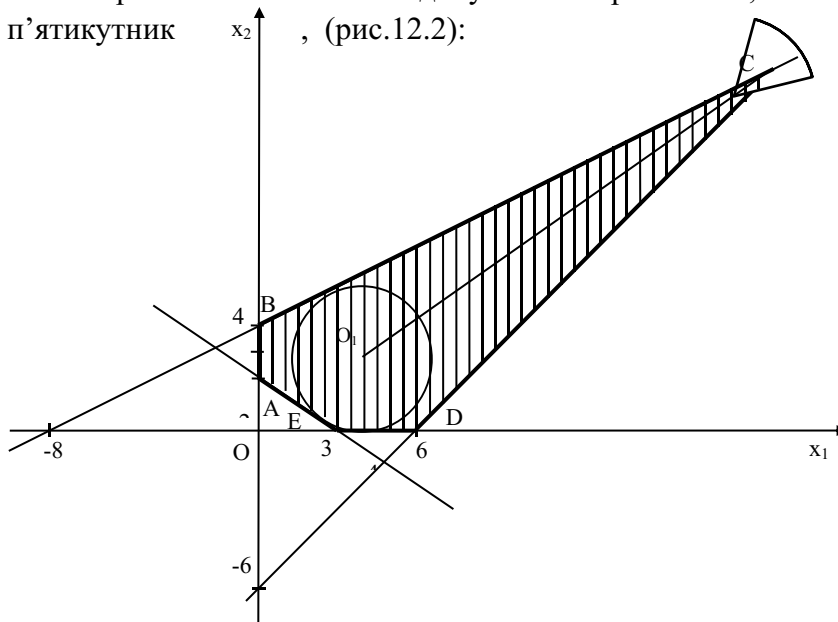


Рисунок 12.2

Лініями рівня функції $f(x,y)$ є лінії $f(x,y) = c$, тобто $f(x,y) = c$ – це концентричні кола з центром O_1 .

Очевидно, що $f(x,y) = 0$, а меншою нуля функція бути не може (як сума квадратів). Так як точка O_1 належить області D , то це і є O_1 . Максимумом функції $f(x,y)$ в області D буде точка, в якій кола $f(x,y) = c$ перетинаються з межею області, причому c , коли радіус кола буде найбільший. З геометричних міркувань видно, що радіус кола $f(x,y) = c$ буде найбільший, коли $c = R^2$. Таким чином, точка O_1 і є точкою, в якій $f(x,y)$ на області D .

Тоді $f(x,y)$ досягне свого максимуму в точці O_1 .

Відповідь: $f(x,y)$ досягне свого максимуму в точці O_1 .

13 Задачі нелінійного програмування без обмежень і з обмеженнями-нерівностями

13.1 Постановка задачі нелінійного програмування

При розв'язуванні задач оптимального планування (або оптимального управління) доводиться враховувати нелінійний характер взаємозв'язків між економічними показниками. У загальному вигляді нелінійна математична модель має вигляд:

при обмеженнях $f(x,y) = c$,

Тоді точки екстремуму для функції
мають вигляд:

(3) – це система алгебраїчних рівнянь. З цієї системи знаходимо розв’язок та . Характер екстремуму визначається за допомогою достатніх умов.

Теорема 1 (про достатні умови екстремуму функції багатьох змінних). Нехай – двічі неперервно диференційовна функція в околі стаціонарної точки

. Тоді точка :

1) є точкою мінімуму функції, якщо

$$\Delta_2 > 0, \quad \Delta_4 > 0, \quad \Delta_6 > 0, \quad \dots, \quad \Delta_n > 0, \quad (4)$$

причому рівність виконується лише за умови ;

2) є точкою максимуму функції, якщо

$$\Delta_2 < 0, \quad \Delta_4 < 0, \quad \Delta_6 < 0, \quad \dots, \quad \Delta_n < 0, \quad (5)$$

причому рівність виконується лише за умови ;

3) не є точкою екстремуму, якщо Δ набуває як додатних, так і від'ємних значень.

Умови 1)-3) означають відповідно, що квадратична форма відносно диференціалів незалежних змінних

Δ є додатно визначена, від'ємно визначена, невизначена.

Нагадаємо, що у вищій алгебрі квадратичну форму

$$Q(x_1, x_2, \dots, x_n) = \sum_{i,j=1}^n a_{ij} x_i x_j, \quad (6)$$

від змінних $x_1, x_2, \dots, x_n$ називають **визначеною додатною (від'ємною)**, якщо вона приймає додатні (від'ємні) значення при всіх значеннях аргументів, що не дорівнюють одночасно нулю.

Відомий критерій Сільвестра, що є **необхідною і достатньою умовою** визначеності і додатності квадратичної форми (6). Цей критерій задається ланцюжком нерівностей:

$$\Delta_1 > 0; \Delta_2 > 0; \dots; \Delta_n > 0 \quad (7)$$

Визначена від'ємна форма із зміною знаку перетворюється на визначену додатну, і навпаки.

Для функції двох змінних форма

при $\Delta > 0$, буде **визначеною** (додатною при $\Delta > 0$ і від'ємною при $\Delta < 0$), а при $\Delta < 0$ – **невизначеною**.

Отже, якщо:

1) $\Delta > 0$, то в стаціонарній точці

функція має екстремум: при - максимум,
 – мінімум;

2) – у точці функція
 не має екстремуму;

3) – сумнівний випадок (потребує
 дослідження з допомогою диференціалів вищих порядків).

Другий диференціал функції багатьох змінних

є симетричною квадратичною формою відносно
 диференціалів незалежних змінних

Означення. Матриця квадратичної форми (8), елементи
 якої є частинні похідні другого порядку функції

, тобто , називається **матрицею Гессе**:

(9)

Визначник матриці H називається **гессіаном**.

У випадку функції двох змінних достатні умови
 екстремуму з використанням гессіана формулюються так:

Теорема 2. Нехай функція двічі неперервно

диференційовна в околі стаціонарної точки . Тоді

точка :

1) є точкою мінімуму, якщо в ній

;

2) є точкою максимуму, якщо в ній

;

3) не є точкою екстремуму, якщо ;

4) потрібне додаткове дослідження за допомогою

диференціалів вищих порядків, якщо .

Приклад 13.1

Знайти точки екстремуму функції

,

при умові
Розв'язання.

Напишемо обмеження у вигляді:

Запишемо функцію Лагранжа:

Запишемо систему для нашої функції Лагранжа:

З другого рівняння системи маємо
 $y = \frac{1}{2}x$, або $y = -\frac{1}{2}x$.
 Якщо $x = 0$, то маємо:

тобто $y = 0$ або $y = 0$.
 Якщо $x \neq 0$, то

звідки або
 Таким чином, наша функція Лагранжа (тобто і дана функція
 $L(x, y, z)$) має чотири стаціонарні точки:

Для визначення типу екстремуму обчислимо матрицю
 Гессе H .

Маємо: $\frac{\partial L}{\partial x} = 0$; $\frac{\partial L}{\partial y} = 0$; $\frac{\partial L}{\partial \lambda} = 0$.

Таким чином: $x = 1$, $y = 1$.

Визначимо знаки мінорів: $\Delta_1 = 1$, $\Delta_2 = 1$.

Як відомо, при $(1, 1)$ є екстремум функції двох змінних, зокрема, при $(1, 1)$ маємо мінімум.

Таким чином, в точках $(1, 1)$ функція L при заданому обмеженні досягає умовного мінімуму.

$\frac{\partial^2 L}{\partial x^2} = 2$; $\frac{\partial^2 L}{\partial y^2} = 2$;

14 Задачі дробово-лінійного програмування

Деякі задачі нелінійного програмування можна звести до задач лінійного програмування. Такою є задача дробово-лінійного програмування.

;

Нехай область допустимих розв'язків непорожня,
тобто система $\begin{cases} a_1x + b_1y \leq c_1 \\ a_2x + b_2y \leq c_2 \end{cases}$ сумісна.

Припустимо, що $a_1 > 0$, $b_1 > 0$.

Можна вважати, що $c_1 > 0$ (інакше знак “-” перенесемо у чисельник).

Введемо нові змінні

;

Тоді $x = x_0 + x_1$, $y = y_0 + y_1$.

(6)

Оскільки $a_1 > 0$ (бо $a_1 > 0$), то з $a_1x_1 + b_1y_1 \leq c_1 - a_1x_0 - b_1y_0$ випливає, що

Кожне рівняння системи (2) помножимо на y_0 :

З умови (4) маємо ще одне обмеження

Отже, задача (1)-(3) зводиться до задачі лінійного програмування (6)-(9), яку можна розв'язати симплекс-методом.

Приклад 14.1 Розв'язати задачу

;

Запишемо задачу в канонічному вигляді

Введемо позначення

;

Маємо задачу , , .

;

Зведемо систему обмежень до базисної форми, виключивши за допомогою третього рівняння з першого та другого рівнянь:

;

Розв'яжемо отриману задачу симплекс-методом

				2	-1	0	0	0
1		0	2	3	2	1	0	0
2		0	6	8	13	0	1	0
3		0	1	1	2	0	0	1
			0	-2	1	0	0	0
1		2		1			0	0
2		0		0			1	0
3		0		0			0	1
				0			0	0

Отже,

;

,

,

Отже, $\dots$, $\dots$.

Відповідь: $\dots$, $\dots$.

Зауваження. Задачу $\dots$ - при $\dots$ можна розв'язати графічно.

Нехай маємо задачу

;

Нехай $c_0=d_0=0$. Область допустимих розв'язків є опуклим многокутником. Вияснимо, як поведуть себе лінії рівня

Розв'яжемо цю рівність відносно $\dots$:

;

Маємо пучок прямих $\dots$, де $\dots$ - довільне число.

Можна довести, що лінії рівня зростають при русі за

годинниковою стрілкою при $\dots$ і проти стрілки годинника при $\dots$.

Якщо ж $\dots$, $\dots$, то лініями рівня є пучок прямих, що проходять через точку $\dots$, де $\dots$ і $\dots$ є розв'язками системи

Приклад 14.2 Розв'язати графічно задачу з *прикладу 14.1*

;

;

Будуємо область допустимих розв'язків (рис.14.1):

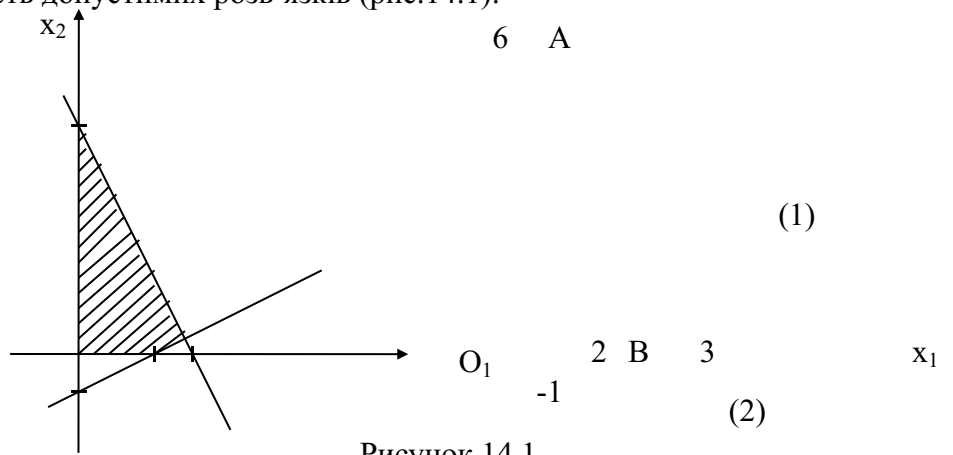


Рисунок 14.1

Знайдемо центр пучка прямих (ліній рівня):

,

,

.

Знайдемо $\frac{d^2f}{dx^2}$, тому значення збільшується за годинниковою стрілкою.

Точка мінімуму – $(-1, -1)$; точка максимуму – $(1, 1)$.

Відповідь: $(-1, -1)$, $(1, 1)$.

15 Задачі опуклого програмування

Означення 1. Функція $f(x)$, яка задана на опуклій множині D , називається **опуклою**, якщо для довільних двох точок x_1 і x_2 і довільного $\lambda \in [0, 1]$, правильна нерівність

Означення 2. Функція $f(x)$ називається **строго опуклою**, якщо в умові знак $\geq$ замінено на $>$ і

Означення 3. Функція $f(x)$ називається **вгнутою** на опуклій множині D , якщо для довільних x_1 і x_2 і довільного $\lambda \in [0, 1]$, виконується:

Формули (1), (2) дають одночасно методи дослідження функції на опуклість. Але часто так досліджувати складно. У ряді випадків дослідження на опуклість спрощується.

Нехай $(x_1, \dots, x_n)$ має неперервні частинні похідні другого порядку. Розглянемо матрицю

де

Матриця називається **матрицею Гессе** функції $(x_1, \dots, x_n)$.

Теорема. Нехай D – опукла множина і $x_0 \in D$. Для того, щоб функція була строго опукла в деякому околі точки необхідно і досить, щоб в цьому околі були додатні всі головні мінори її матриці. Якщо всі непарні головні мінори від’ємні, а парні – додатні, то функція строго вгнута в околі точки.

Якщо наведені умови опуклості виконуються для всіх $x_0 \in D$, то функція є строго опуклою скрізь на D .

Приклад 15.1 Вивчити поведінку функції

в околі точки

Розв’язання.

– від не залежить.

Отже, строго вгнута для всіх f .
 Основною властивістю опуклих функцій є те, що їх локальний екстремум є одночасно і глобальним. Сформулюємо точно ці результати.

1. Якщо f – опукла (вгнута) функція, задана на замкненій опуклій множині S , то всякий локальний мінімум (максимум) функції на області S є глобальним.
2. Якщо глобальний мінімум (максимум) досягається у двох різних точках, то він досягається у кожній точці відрізка, що з'єднує дані точки.
3. Якщо f – строго опукла (вгнута) функція, то її глобальний мінімум (максимум) на опуклій множині S досягається в єдиній точці.
4. Нехай f – опукла (вгнута) на опуклій множині S , f – має неперервні похідні першого порядку всередині S . Нехай x^* – стаціонарна точка f . Тоді x^* досягає в точці x^* глобального мінімуму (максимуму).

Ці результати дають змогу розв'язувати задачі нелінійного програмування у випадку коли f – опукла (вгнута), множина допустимих розв'язків S опукла. Якщо при цьому f має неперервні частинні похідні першого порядку в x^* , то для знаходження мінімуму (максимуму) досить знайти стаціонарну точку функції f .

Розглянемо задачу опуклого програмування з обмеженнями-нерівностями виду

;

де функції f , g_i – опуклі вгору.
 Нехай задача (1)-(3) задовольняє умову регулярності
 Слейтера: існує хоч одна точка x^0 , в якій

Метод множників Лагранжа можна поширити і на задачі (1)-(3), де обмеження задані нерівностями.

Означення 4. Функцією Лагранжа задачі (1)-(3) називають

Означення 5. Точка x^* називається сідловою точкою Лагранжа, якщо для всіх

Теорема Куна-Таккера. Нехай задача (1)-(3) задовольняє умову регулярності Слейтера, функції f , g_i є опуклими вгору. Вектор λ^* є оптимальним розв'язком задачі - тоді і тільки тоді, коли існує такий вектор λ^* , що x^* є сідловою точкою функції Лагранжа $L(x, \lambda^*)$.

Отже, задача опуклого програмування (1)-(3) зводиться до задачі знаходження сідлових точок функції Лагранжа: треба знайти точки, в яких функція Лагранжа досягає максимуму по λ і мінімуму по x . Ця задача називається *задачею максимінімізації* і може бути записана так:

Якщо функції f , g_i диференційовані, то можна довести, що необхідними і додатними умовами того, що

– сідлова, є такі:

$$, \quad , \quad , \quad ;$$

Зауваження. Ці умови називаються локальними умовами Куна-Таккера (умовами ортогональності пари двоїстих задач лінійного програмування). По аналогії і у нелінійному програмуванні говорять про двоїсті задачі, причому одна з них

,

друга:

де S – множина допустимих розв'язків, визначена системою (2),(3).

Використовуючи ці результати, можна розглянути задачі нелінійного програмування у найзагальнішому вигляді:

;

Зведемо всі обмеження до одного типу:

Складемо функцію Лагранжа

Знайдемо її частинні похідні:

$$\begin{aligned} \frac{\partial L}{\partial x_i} &= 0, & \frac{\partial L}{\partial y_j} &= 0; \\ \frac{\partial L}{\partial \lambda_k} &= 0, & \frac{\partial L}{\partial \mu_l} &= 0; \end{aligned}$$

У цьому випадку умови теореми Куна-Таккера формуються так. Нехай f, g_i, h_j - диференційовані функції. Для того, щоб вектор x^* був оптимальним розв'язком задачі, необхідно, щоб існував вектор (λ^*, μ^*) такий, що виконуються умови:

- 1) $\lambda_k \geq 0, \mu_l \geq 0$;
- 2) $\lambda_k g_k(x^*) = 0, \mu_l h_l(x^*) = 0$;
- 3) $\lambda_k g_k(x^*) \leq 0, \mu_l h_l(x^*) = 0$, причому якщо $\lambda_k > 0$, то $g_k(x^*) = 0$, якщо $\mu_l > 0$, то $h_l(x^*) = 0$.

Якщо точка x^* задовольняє ці умови, функція $f(x^*)$ є

опуклою вверх, λ_1 – опуклі вверх, λ_2 – опуклі
вниз, то (x^*, y^*) є сідловою точкою функції Лагранжа $L(x, y, \lambda)$, а
 (x^*, y^*) є оптимальним розв'язком задачі нелінійного
програмування.

Приклад 15.1 Розв'язати задачу опуклого програмування

Розглянемо функцію

Запишемо функцію Лагранжа

Запишемо умови Куна- Таккера

Розв'яжемо задачу методом штучного базису:

i	B	C B	B	0	0	0	0	0	0	0	0	0	0	- M
				A_{x_1}	A_{x_2}	A_{y_1}	A_{y_2}	A_{y_3}	A_{v_1}	A_{v_2}	A_{w_1}	A_{w_2}	A_{w_3}	A_{y_1}
1	A_{v_1}	0	8	- 8	0	-1	-2	-1	1	0	0	0	0	0

2	A_{yl}	$\bar{M}$	2	0	8	1	1	2	0	-1	0	0	0	1
3	A_{wl}	0	1	1	1	0	0	0	0	0	1	0	0	0
4	A_{w2}	0	4	2	2	0	0	0	0	0	0	1	0	0
5	A_{w3}	0	6	1	2	0	0	0	0	0	0	0	1	0
m_2^+			-2	0	$\bar{8}$	-1	-1	-2	0	1	0	0	0	$\bar{1}$
1	A_{v1}	0	8	$\bar{8}$	0	-1	-2	-1	1	0	0	0	0	0
2	A_{x2}	0		0	1				0		0	0	0	
3	A_{wl}	0		1	0				0		1	0	0	0
4	A_{w2}	0		2	0				0		0	1	0	0
5	A_{w3}	0		1	0				0		0	0	1	0
m_1^+			0	0	0	0	0	0	0	0	0	0	0	0

Перевіримо умови Куна-Таккера:

$$0 \cdot 8 = 0 \quad 0 \cdot 0 = 0 \quad 0 \cdot 0 = 0 \quad 0 \cdot 0 = 0 \quad 0 \cdot 0 = 0$$

Отже, сідлова точка

Тоді оптимальний розв'язок:

16 Елементи динамічного програмування

У задачах лінійного і нелінійного програмування економічний процес вважався стаціонарним (не залежним від часу). Насправді економіка динамічна, тому щоб повніше відобразити економічні процеси, треба шукати розв'язки не для одного проміжку часу, а на протязі кількох етапів. Наприклад, планувати роботу підприємства краще не на рік, а по місяцях, враховуючи результати попереднього місяця.

Тому виникають задачі *динамічного проектування*, тобто задачі, у яких на кожному етапі приймаються оптимальні рішення. Ці рішення визначають оптимальну стратегію поведінки. Причому рішення на кожному етапі приймається виходячи з інтересів усієї задачі, тому вони можуть не бути оптимальними в рамках одного етапу.

Розглянемо метод розв'язання задач динамічного програмування на прикладі задачі про оптимальний розподіл капіталовкладень (або ресурсів).

Нехай у два підприємства і вкладають капіталовкладення у розмірі k грошових одиниць. Нехай першому підприємству виділено x грошових одиниць, тоді другому $-k-x$. Прибуток від буде , від $-$. Тоді сумарний прибуток буде

Він має бути максимальним:

Нехай залишок капіталовкладень після першого етапу на першому підприємстві Π_1 становитиме , а на

другому , де . Тоді прибуток за другий етап складе

Сумарний максимальний прибуток за два етапи буде

Якщо - максимальний прибуток за три етапи, то

Аналогічно для етапів одержимо

де – найбільший прибуток за попередні етапів.
Одержали рекурентне співвідношення, яке називається **функціональним рівнянням Беллмана**.

Рівняння означає, що максимальний прибуток виробництва від -етапного процесу одержується як максимум суми прибутків від першого етапу і наступних етапів при умові, що залишені після першого етапу ресурси протягом наступних етапів використовуються найкращим чином.

Рівняння (1) є математичним виразом **принципу Беллмана**: якщо деяка послідовність розв'язків оптимальна, то окремі наступні розв'язки оптимальні по відношенню до попередніх.

У відповідності з принципом Беллмана задачу розв'язують, починаючи з останнього етапу. Потому переходять до розв'язання попереднього етапу, відкидаючи можливості альтернативи, які протирічать вибраному розв'язку.

Пропонуємо розв'язання задачі динамічного програмування за допомогою принципу Беллмана на прикладі.

Приклад 16.1 Розподілити капіталовкладення $K=200$ тис. грн. чотирьом підприємствам, якщо відомі їх функції приросту випуску продукції , , , , задані для деяких значень таблицею.

Загальний приріст випуску продукції має бути максимальний.

Капіталовкладення , тис. грн	Приріст випуску продукції (прибуток)			
0	0	0	0	0
50	25	30	36	28
100	60	70	64	56
150	100	90	95	110
200	140	122	130	142

Позначимо , — максимальний приріст випуску продукції при розподілі капіталовкладень тис. грн. відповідно між першим, двома першими, трьома першими, чотирма заводами.

Функціональне рівняння:

Оскільки , то

,

,

,

,

.

Оскільки , то

Оскільки , то ;

Оскільки , то ;

Одержані дані занесемо у таблицю

Капіталовкладення,				
50	25	30	36	36
100	60	70	70	70

150	100	100	106	110
200	140	140	140	146

За таблицею максимальний приріст випуску продукції 146 тис. грн. при капіталовкладенні 200 тис. грн. Він досягається, якщо четвертому заводу надати 150 тис. грн., а трьом першим 50 тис. грн.; 50 тис. грн. розподіляються так – всі 50 тис. грн. третьому, а двом першим – нічого.

Відповідь: 1-му заводу – 0; 3-му заводу – 50 тис. грн.;
2-му заводу – 0; 4-му заводу – 150 тис. грн.

При цьому максимальний приріст продукції складе 146 одиниць.

17 Елементи сіткового планування

17.1 Поняття про сіткове планування

Методи сіткового планування використовують у тих галузях, які пов'язані з плануванням, аналізом і здійсненням великої кількості операцій. Ефективність методів сіткового планування виявляється в наступному: скорочення термінів виконання робіт, чітке врахування всіх робіт за часом, встановлення кінцевого ланцюжка робіт, можливість оперативного втручання в процес розробок, регулювання розроблених планів, тісний зв'язок всього колективу робіт, які беруть участь у процесі, підлеглість його центральним органам.

У сітковому плануванні розглядаються об'єкти двох видів: події і операції. **Подія** – це стан системи в момент досягнення деякої вихідної проміжної або кінцевої мети. **Операція** – сама робота, результатом якої є подія.

Великий комплекс майбутніх операцій доцільно представити за допомогою так званого графа.

Використаємо зображення

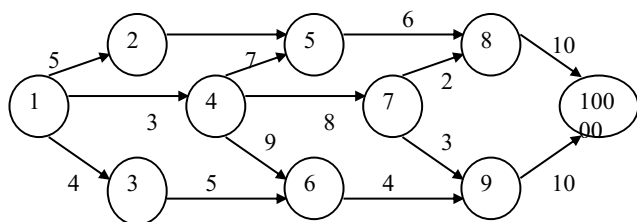


$\bigcirc i, \bigcirc j$ – події. Подія $\bigcirc i$ передуює $\bigcirc j$ або $\bigcirc j$ слідує за $\bigcirc i$.

Тривалість роботи дорівнює
Нехай при плануванні деякої розробки виділено 15 робіт, причому для кожної вказано попередню і наступну події та тривалість кожної роботи

Робота		I	II	III	IV	V	VI	VII	VIII	IX	X	XI	XII	XIII	XIV	XV
Події	Попередня	1	1	1	2	3	3	4	4	4	5	6	7	7		
	Наступна	2	3	4	5	4	6	5	6	7	8	9	8	9		
Тривалість		5	4	3	1	3	5	7	9	8	6	4	2	3		

За таблицею будемо відповідну схему послідовного виконання робіт і вказуємо тривалості робіт:



Така схема називається *графом*. Згідно з математичною теорією графів, повинні виконуватись умови:

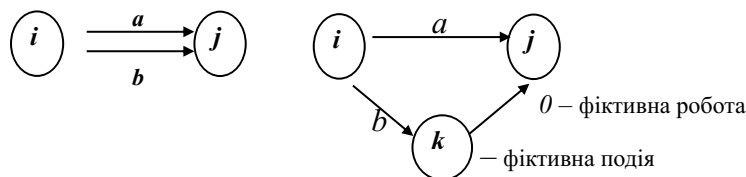
- 1) з початкового кружка стрілки тільки виходять;
- 2) в останній кружок стрілки тільки входять;
- 3) у проміжних кружків є і вхідні, і вихідні стрілки.

Таким чином, *граф* – це схема, яка складається з точок і ліній, що є відповідно прототипами подій і операцій. Точки – вершини графа, події – його ребра, або дуги. Розрізняють орієнтований або неорієнтований графи. Граф називається *орієнтованим*, якщо вказано стрілками напрямки всіх його ребер. *Шляхом* називається послідовність ребер графа, в якій кінець кожного попереднього співпадає з початком наступного. *Контуром* називається деякий кінцевий шлях, у

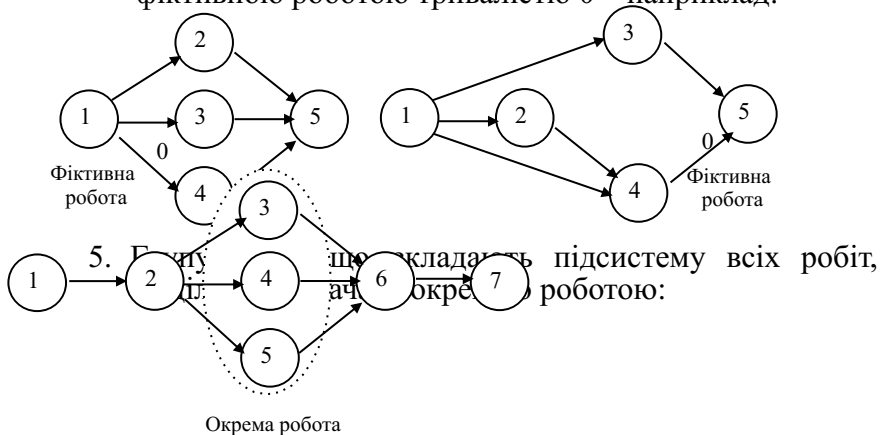
якому початок співпадає з кінцем. **Сітковим графом** називається орієнтований без контурів граф, ребра якого мають одну чи декілька числових характеристик.

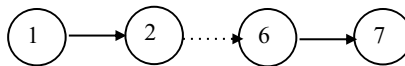
Правило побудови сіткового графа:

1. Кожен сітковий граф має лише одну початкову вершину, з якої тільки виходять орієнтовані дуги (стрілки).
2. Кінцева вершина графа – вершина, в яку тільки входять стрілки. У проміжних вершин є вхідні і вихідні стрілки.
3. Дві послідовні події мають з'єднуватися тільки однією стрілкою. Якщо є дві паралельні роботи, то вводиться фіктивна подія і фіктивна робота тривалістю 0:



4. У сітковому графі не повинно бути зависань, тобто всі події (крім першої і останньої) повинні мати і вхідні, і вихідні стрілки. Якщо такого немає, то їх з'єднують з початковою або кінцевою подією фіктивною роботою тривалістю 0 – наприклад:





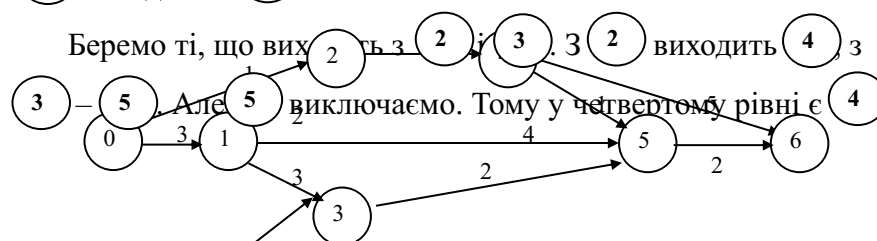
6. Якщо $i < j$, то подія i передуює події j .

Приклад 17.1. Скласти граф, якщо задані вершини і відстані між вершинами

Вершини	Попередня	0	0	1	1	1	2	3	4	4	5
	Наступна	1	2	2	3	5	4	5	5	6	6
Відстані		3	1	2	3	4	3	2	1	5	2

Спочатку вибираємо подію, яка не є наступною для жодної, це 0 . Вона складає 1-й рівень. Далі беремо ті, що виходять з 0 . Це 1 і 2 . З них вибираємо ті, що не містять вхідних стрілок. Для 2 є вхідна з 1 , тому 2 не беремо до уваги. Залишається 1 , що складає другий рівень.

Беремо ті, що виходять з 1 . Це $2, 3, 5$. Оскільки з 2 і 3 немає вихідних в $0, 1, 2, 3, 5$, то вони входять у третій рівень. В третій рівень не входить 5 , бо в 5 є вхідний з 3 .



З (4) виходить (5) і (6), але (6) виключаємо. У 5-му рівні (5). На 6-му – (6) добавляємо зв'язки (0) → (2), (1) → (5), (3) → (5), (4) → (6).

Одержимо орієнтований граф.

5.2 Алгоритм пошуку найкоротшого шляху мережі

Часто потрібно визначити найкоротший шлях між містами, найменшу тривалість послідовності робіт тощо. Розглянемо алгоритм пошуку найкоротшого шляху на попередньому прикладі.

У алгоритмі перегляд вершин проводиться в послідовності зменшення їх номерів. Біля кружка записуємо найкоротшу довжину шляху від нього до кінцевого кружка. Найкоротший шлях позначаємо подвійною стрілкою.

Рухаємось від кружка з останнього рівня, тобто від (6). Ставимо 0 біля (6), бо звідси відраховуємо шлях.

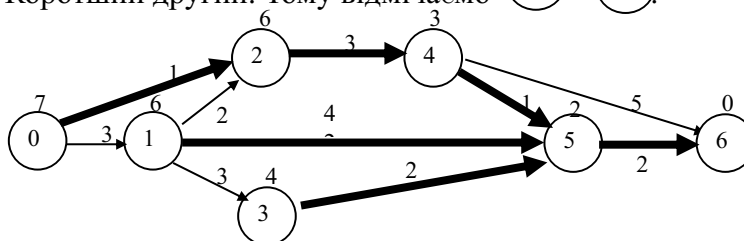
Переходимо до кружків у п'ятому рівні. Такий один – (5) тому шлях єдиний. Позначимо (5) (6) і відстань 2.

Переходимо до кружків у четвертому рівні, це (4).

Є два переходи до останнього:

(4) (6) з відстанню 5 і (4) (5) (6) з відстанню 1+2=3.

Коротший другий. Тому відмічаємо (4) (5).



Далі з третього рівня, тобто, до (2) і (3).

Шлях до (6) може бути такий:

(2) (4) (5) (6) з відстанню $3+3=6$, або

(3) (5) (6) з відстанню $2+2=4$.

Найменше 4, тому позначаємо (3) (5).

Переходимо до другого рівня, тобто до (2).

З (1) до (6) є такі шляхи:

(1) (2) (4) (5) (6) з відстанню $2+6=8$;

(1) (5) (6) з відстанню $4+2=6$;

(1) (3) (5) (6) з відстанню $3+4=7$.

Найкоротший другий, тому позначимо (1) (5).

Переходимо на перший рівень. До (6) є шляхи:

(0) (2) (4) (5) (6) з відстанню $1+6=7$;

(0) (1) (5) (6) з відстанню $3+6=9$;

(0) (1) (3) (5) (6) з відстанню $3+7=10$.

Отже, найкоротший шлях (0) (2) (4) (5) (6), а найменша відстань – 7.

З даного прикладу ми можемо визначити критичний шлях, який з'єднує початкову і кінцеву вершини і має найдовшу тривалість 10:

Перейдемо до обчислення часових характеристик

сіткового графа: раннього і пізнього терміну виконання робіт і резерву часу:

–ранній термін , де ;

–пізній термін ;

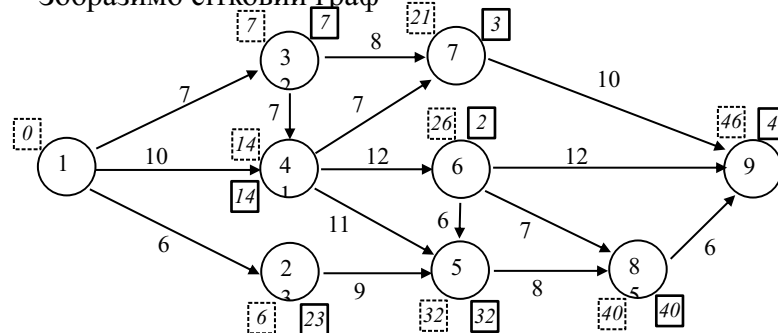
–резерв часу: .

Приклад 17.2

Побудувати сітковий граф і знайти його часові характеристики.

Вершини	А	Б	В	Г	Д	Е	Є	Ж	З	И	І	Ї	Й	К	Л
Попередня	1	1	1	2	3	3	4	4	4	7	6	6	6	5	8
Наступна	2	3	4	5	7	4	7	6	5	9	9	5	8	8	9
Тривалість	6	7	10	9	8	7	7	12	11	10	12	6	7	8	6

Зобразимо сітковий граф



Визначимо ранній термін виконання робіт:

Визначимо пізній термін виконання робіт:

Тоді резерв часу:

Зауваження. Застосовувати методи сіткового планування можна і для розв'язування транспортної задачі, при цьому виконання робіт – це тривалість здійснення перевезень або відстані між відповідними населеними пунктами.