

К. П. Барахов, С. С. Курєннов

МЕТОДИ ЛІНІЙНОЇ ТА ЦІЛОЧИСЛОВОЇ ОПТИМІЗАЦІЇ

2019

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний аерокосмічний університет ім. М. Є. Жуковського
«Харківський авіаційний інститут»

К. П. Барахов, С. С. Курєннов

МЕТОДИ ЛІНІЙНОЇ ТА ЦІЛОЧИСЛОВОЇ ОПТИМІЗАЦІЇ

Навчальний посібник

Харків «ХАІ» 2019

УДК 519.8
Б58

Рецензенти: д-р фіз.-мат. наук, проф. Н. В. Сметанкіна,
д-р фіз.-мат. наук, проф. О. О. Стрельнікова

Барахов, К. П.

Б58 Методи лінійної та цілочислової оптимізації [Електронний ресурс] : навч. посіб. / К. П. Барахов, С. С. Курєннов. – Харків : Нац. аерокосм. ун-т ім. М. Є. Жуковського «Харків. авіац. ін-т», 2019. – 64 с.

Викладено основи математичного програмування - науки, що досліджує оптимізаційні методи, які є кількісним обґрунтуванням оптимальних управлінських рішень в економіці. Розглянуто основні поняття і методологічні принципи математичного програмування (лінійне, дробово-лінійне, цілочислове), математичні методи оптимізації.

Матеріал подано на рівні, доступному студентам, які знайомі з курсом вищої математики. Методи, що розглянуто в посібнику, ілюструються значною кількістю прикладів. Мета посібника – навчити студентів застосовувати математику та обчислювальну техніку для обґрунтування оптимальних економічних рішень.

Для студентів спеціальностей таких галузей знань: 05 «Соціальні та поведінкові науки», 07 «Управління та адміністрування», 11 «Математика та статистика», 12 «Інформаційні технології».

Іл. 9. Табл. 33. Бібліогр.: 12 назв

УДК 519.8

© Барахов К. П., Курєннов С. С., 2019
© Національний аерокосмічний
університет ім. М. Є. Жуковського
«Харківський авіаційний інститут», 2019

Зміст

1. Лінійне програмування.....	4
1.1. Канонічні форми лінійних оптимізаційних моделей.....	4
1.2. Перетворення задач.....	4
1.3. Симплекс-алгоритм.....	6
1.4. Метод штучного базису.....	10
1.5. Графічна інтерпретація симплекс-методу.....	11
1.6. Аналіз моделей на чутливість.....	16
1.7. Двоїстість.....	19
1.8. Двоїстий симплекс-алгоритм.....	24
1.9. Збіжність симплекс-алгоритму.....	26
2. Цілочислове програмування.....	28
2.1. Задачі цілочислового програмування.....	28
2.2. Метод Гоморі.....	30
2.3. Метод гілок і меж.....	33
2.4. Задача про призначення.....	37
2.5. Задача комівояжера.....	38
3. Транспортна задача.....	42
3.1. Постановка транспортної задачі.....	42
3.2. Властивості закритої транспортної задачі.....	43
3.3. Властивості опорних планів транспортної задачі.....	44
3.4. Алгоритми побудови опорних планів транспортної задачі.....	48
3.5. Критерій оптимальності плану перевезень.....	52
3.6. Алгоритм методу потенціалів.....	53
3.7. Приклади задач, які зводяться до транспортної задачі.....	57
3.8. Транспортна задача за критерієм часу.....	61
Бібліографічний список.....	63

1. ЛІНІЙНЕ ПРОГРАМУВАННЯ

1.1. Канонічні форми лінійних оптимізаційних моделей

Будь-яку лінійну оптимізаційну модель зручно подати у компактному стандартному і цілком детермінованому вигляді. Перетворення лінійних рівнянь і нерівностей, які будуть наведені нижче, дозволяють розв'язати цю задачу. На даний час існує велика кількість різновидів канонічних форм. Розглянемо дві найбільш поширені форми.

Будь-яку задачу лінійного програмування можна розглядати як задачу максимізації:

$$L = \sum_{j=1}^n c_j x_j \rightarrow \max \quad (1.1)$$

за наявності обмежень

$$\sum_{j=1}^n a_{i,j} x_j \leq b_i, \quad (i=1, 2, \dots, m), \quad (1.2)$$
$$x_j \geq 0, \quad (j=1, 2, \dots, n).$$

У той же час будь-яку задачу лінійного програмування можна звести до задачі мінімізації:

$$L = \sum_{j=1}^n c_j x_j \rightarrow \min \quad (1.3)$$

за наявності обмежень

$$\sum_{j=1}^n a_{i,j} x_j = b_i, \quad (i=1, 2, \dots, m), \quad b_i \geq 0, \quad (1.4)$$
$$x_j \geq 0, \quad (j=1, 2, \dots, n).$$

Як правило (хоча це не є обов'язковим), у рівнянні (1.4) має місце нерівність $n > m$.

1.2. Перетворення задач

Перехід від максимізації до мінімізації. В лінійному програмуванні будь-яка задача максимізації може бути перетворена до еквівалентної задачі мінімізації, і навпаки, якщо одночасно зі зміненням «знака» оптимізації змінити знаки перед усіма коефіцієнтами у виразах для цільової функції (1.1) або (1.4). Так, наприклад, максимізація $\sum_{j=1}^n c_j x_j$

еквівалентна мінімізації $\sum_{j=1}^n (-c_j) x_j$. Якщо при цьому V - оптимальне

значення лінійної форми $\sum_{j=1}^n (-c_j) x_j$, то $-V$ являє собою оптимальне значення лінійної форми $\sum_{j=1}^n c_j x_j$.

Перехід до еквівалентної системи нерівностей. Кожне з нерівностей, що фігурують у моделі лінійного програмування, можна записати в інверсійній формі, якщо врахувати, що

$$\sum_{j=1}^n a_{i,j} x_j \leq b_i \text{ еквівалентно } \sum_{j=1}^n (-a_{i,j}) x_j \geq -b_i.$$

Обернення нерівності у рівність. Будь-яку нерівність у моделі можна перетворити у рівність, якщо додати нову невід'ємну змінну:

$$\sum_{j=1}^n a_{i,j} x_j \leq b_i \text{ можна записати як } \sum_{j=1}^n a_{i,j} x_j + x_{j+1} = b_i, \text{ де } x_{j+1} \geq 0;$$

$$\sum_{j=1}^n a_{i,j} x_j \geq b_i \text{ можна записати як } \sum_{j=1}^n a_{i,j} x_j - x_{j+1} = b_i, \text{ де } x_{j+1} \leq 0.$$

Нові введені в модель змінні зазвичай мають назву «вільні змінні» та можуть бути інтерпретовані, наприклад, як залишок товару, місця на складі, оборотні кошти і т.п., або, навпаки, як перевищення попиту над наявним товаром.

Обернення рівності у нерівність. Будь-яке лінійне рівняння, а також систему лінійних рівнянь можна подати у вигляді деякої сукупності лінійних нерівностей за допомогою введення одного додаткового обмеження. Систему рівнянь $\sum_{j=1}^n a_{i,j} x_j = b_i$, $(i=1, 2, \dots, m)$ можна подати у

вигляді сукупності нерівностей $\sum_{j=1}^n a_{i,j} x_j \leq b_i$, $(i=1, 2, \dots, m)$ і ще однієї

додаткової нерівності $\sum_{j=1}^n \alpha_j x_j \leq \beta$, де $\alpha_j = -\sum_{i=1}^m a_{i,j}$, $\beta = -\sum_{i=1}^m b_i$.

Перехід від змінних, які не мають обмеження на знак, до невід'ємних змінних. Змінна x_j , яка не має обмежень на знак, вважається такою, що дорівнює різниці двох невід'ємних змінних: $x_j = x'_j - x''_j$. І цю різницю підставляють замість змінної x_j там, де вона фігурує. Ця підстановка збільшує кількість змінних у моделі, зберігаючи лінійність.

Перехід від змінних, які мають обмеження знизу, до невід'ємних змінних. Якщо змінна x_j обмежена знизу деякою константою β_j , можливо зробити перехід до нового формулювання задачі лінійного

програмування, де замість змінної x_j фігурує невід'ємна змінна y_j , яка пов'язана з x_j перетворенням $x_j = \beta_j + y_j$, де $y_j \geq 0$.

1.3. Симплекс-алгоритм

Для розв'язання задач лінійного програмування при довільному числі змінних x_j застосовуються обчислювальні методи, найбільш універсальним серед яких є симплексний метод.

Розглянемо застосування цього методу на такому прикладі.

Приклад 1.1. Знайти максимум цільової функції

$$L = 4x_1 + 5x_2 + 9x_3 + 11x_4$$

при обмеженнях

$$\begin{cases} x_1 + x_2 + x_3 + x_4 \leq 15, \\ 7x_1 + 5x_2 + 3x_3 + 2x_4 \leq 120, \\ 3x_1 + 5x_2 + 10x_3 + 15x_4 \leq 100, \\ x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0. \end{cases}$$

Позначимо цільову функцію x_0 і перетворимо нерівності в обмеження у рівності, додавши до них, відповідно, додаткові змінні x_5 , x_6 , x_7 . У результаті отримуємо таку систему рівнянь:

$$\begin{aligned} x_0 - 4x_1 - 5x_2 - 9x_3 - 11x_4 &= 0 \\ x_1 + x_2 + x_3 + x_4 + x_5 &= 15, \\ 7x_1 + 5x_2 + 3x_3 + 2x_4 + x_6 &= 120, \\ 3x_1 + 5x_2 + 10x_3 + 15x_4 + x_7 &= 100, \end{aligned} \tag{1.5}$$

де всі змінні можуть приймати лише невід'ємні значення. Введення змінної x_0 дозволяє записати вираз для цільової функції у вигляді рівняння.

Розглянемо покрокове виконання симплексного методу.

На першому кроці вибирають початковий припустимий розв'язок системи рівнянь (1.5). Існує безліч таких розв'язків, але зручніше усього почати з $x_0 = 0$, $x_5 = 15$, $x_6 = 120$, $x_7 = 100$ при нульових значеннях інших змінних. Іншими словами, перший розв'язок будується за допомогою тільки вільних змінних. Цей розв'язок має назву – початковий базисний розв'язок, або початковий опорний план. Змінні x_0 , x_5 , x_6 , x_7 називаються базисом, а інші змінні – небазисними або позабазисними.

Базис має розмірність, яка дорівнює максимальному числу лінійно незалежних рівнянь у системі (1.5). У нашому випадку це чотири рівняння, і тому маємо чотири базисні змінні.

Крок 2 симплексного алгоритму дозволяє визначити змінну, яка увійде до чергового базису, що дозволить збільшити значення цільові

функції x_0 . Для визначення змінної, що повинна увійти до базису, застосовують симплекс-критерій I (для максимізації).

Симплекс-критерій I (для максимізації). Якщо в першому рядку системи є небазисні змінні, коефіцієнти при яких від'ємні, потрібно вибрати змінну з найбільшим абсолютним значенням коефіцієнта, що стоїть перед нею, тобто ту змінну, що забезпечує найбільше питоме збільшення значення цільової функції. Якщо всі небазисні змінні в першому рядку мають додатні або нульові коефіцієнти, то оптимальний розв'язок отримано.

Таким чином, необхідно ввести до базису змінну x_4 . Збільшення x_4 можливо лише за рахунок зменшення базисних змінних у кожному рівнянні (1.5), що містить x_4 із додатним коефіцієнтом. Змінну x_4 можна збільшувати доти, поки одна з базисних змінних, що зменшується, досягне своєї нижньої межі, тобто нуля. Для отримання значення змінної, яка вийде з базису, та значення змінної, яка увійде до базису, застосовують симплекс-критерій II.

Симплекс-критерій II. Розглянемо відношення значень базисних змінних до відповідних коефіцієнтів при новій базисній змінній (крім нульових і від'ємних коефіцієнтів). Вибираємо найменше відношення. Дорівнюємо нову базисну змінну цьому відношенню. Змінна, що відповідає найменшому відношенню, виходить із базису.

Результати обчислень наведено у табл. 1.1.

Таблиця 1.1

Базисні змінні	Базисний розв'язок	Коефіцієнти при x_4	Значення відношень	Мінімальне значення
x_0	0	-11	-	
x_5	15	1	15	
x_6	120	2	60	
x_7	100	15	6,67	6,67

У новий базис увійде $x_4 = 20/3$, а x_7 вийде з базису і набуде нульового значення. Виключимо x_4 зі всіх рівнянь системи (1.5):

$$\begin{aligned}
x_0 - \frac{9}{5}x_1 - \frac{4}{3}x_2 - \frac{5}{3}x_3 + \frac{11}{15}x_7 &= \frac{220}{3}, \\
\frac{4}{5}x_1 + \frac{2}{3}x_2 + \frac{1}{3}x_3 + x_5 - \frac{1}{15}x_7 &= \frac{25}{3}, \\
\frac{33}{5}x_1 + \frac{13}{3}x_2 + \frac{5}{3}x_3 + x_6 - \frac{2}{15}x_7 &= \frac{320}{3}, \\
\frac{1}{5}x_1 + \frac{1}{3}x_2 + \frac{2}{3}x_3 + x_4 + \frac{1}{15}x_7 &= \frac{20}{3}.
\end{aligned}$$

На наступній ітерації за симплекс-критерієм I необхідно включити до базису x_1 . Симплекс-критерій II вказує, що x_1 залишається у другому рядку, а з базису виключається x_5 . Тобто система набуває вигляду:

$$\begin{aligned}
x_0 + \frac{1}{6}x_2 - \frac{11}{12}x_3 + \frac{9}{4}x_5 + \frac{7}{12}x_7 &= \frac{1105}{12}, \\
x_1 + \frac{5}{6}x_2 + \frac{6}{12}x_3 + \frac{5}{4}x_5 - \frac{1}{12}x_7 &= \frac{125}{12}, \\
-\frac{7}{6}x_2 - \frac{13}{12}x_3 - \frac{33}{4}x_5 + x_6 + \frac{5}{12}x_7 &= \frac{455}{12}, \\
\frac{1}{6}x_2 + \frac{7}{12}x_3 + x_4 - \frac{1}{4}x_5 + \frac{1}{12}x_7 &= \frac{55}{12}.
\end{aligned}$$

На наступній ітерації маємо:

$$\begin{aligned}
x_0 + \frac{3}{7}x_2 + \frac{11}{7}x_4 + \frac{13}{7}x_5 + \frac{5}{7}x_7 &= \frac{695}{7}, \\
x_1 + \frac{5}{7}x_2 - \frac{5}{7}x_4 + \frac{10}{7}x_5 - \frac{1}{7}x_7 &= \frac{50}{7}, \\
-\frac{6}{7}x_2 + \frac{13}{7}x_4 - \frac{61}{7}x_5 + x_6 + \frac{4}{7}x_7 &= \frac{325}{7}, \\
\frac{2}{7}x_2 + x_3 + \frac{12}{7}x_4 - \frac{3}{7}x_5 + \frac{1}{7}x_7 &= \frac{55}{7}.
\end{aligned} \tag{1.6}$$

У першому рядку всі коефіцієнти при небазисних змінних є додатними, тобто розв'язок задачі отримано. Оптимальний розв'язок має вигляд: $x_1 = \frac{50}{7}$, $x_2 = 0$, $x_3 = \frac{55}{7}$, $x_4 = 0$, $x_5 = 0$, $x_6 = \frac{325}{7}$, $x_7 = 0$. При цьому

змінні x_5 , x_6 , x_7 є додатковими. Цільова функція $L = x_0 = \frac{695}{7}$.

Алгоритм симплексного методу

Крок 1. Знайти початковий припустимий базисний розв'язок.

Крок 2. Застосувати симплекс-критерій 1. Якщо оптимальний розв'язок досягнуто, припинити розрахунки, інакше перейти до кроку 3 .

Крок 3. Застосувати симплекс-критерій 2.

Крок 4. Виконати заміну базису .

Крок 5. Перейти до кроку 2.

Зазвичай обчислення за симплекс-методом виконують у вигляді симплекс-таблиці 1.2. У частині таблиці, що відповідає ітерації 0, переписані коефіцієнти системи (1.5), базисні змінні та їхні значення. Стовець табл. 1.2 , якому відповідає змінна, що вводиться до базису, називається *розв'язувальним стовпцем*. Рядок цієї таблиці, якому відповідає змінна, що виводиться з базису, називається – *розв'язувальним рядком*. Елемент, який знаходиться на перетинанні розв'язувального рядка і розв'язувального стовпця, називається *розв'язувальним елементом*.

Для ітерації 1 розв'язувальний стовець – x_4 , розв'язувальний рядок – x_7 , розв'язувальний елемент дорівнює 15. Для наочності нульові значення коефіцієнтів у таблиці не показані, відповідні елементи таблиці є порожніми.

Таблиця 1.2

N	Базис	Базисний розв'язок	x_1	x_2	x_3	x_4	x_5	x_6	x_7
1	x_0	0	-4	-5	-9	-11			
	x_5	15	1	1	1	1	1		
	x_6	120	7	5	3	2		1	
	x_7	100	3	5	10	15			1
2	x_0	220/3	-9/5	-4/3	-5/3				11/15
	x_5	25/3	4/5	2/3	1/3		1		-1/15
	x_6	320/3	33/5	13/3	5/3			1	-2/15
	x_4	20/3	1/5	1/3	2/3	1			1/15
3	x_0	1105/12		1/6	-11/12		9/4		7/12
	x_1		1	5/6	5/12		5/4		-1/12
	x_6			-7/6	-13/12		-33/4	1	5/12
	x_4			1/6	7/12	1	-1/4		1/12
4	x_0	695/7		3/7		11/7	13/7		5/7
	x_1	50/7		5/7		-5/7	10/7		-1/7
	x_6	325/7		-6/7		13/7	-61/7	1	4/7
	x_3	55/7		2/7	1	12/7	-3/7		1/7

Затемненим показаний розв'язувальний рядок та обведено розв'язувальний стовпець. На кожній ітерації розв'язувальний рядок необхідно розділити на коефіцієнт при змінній, яка вводиться до базису, та виключити всі інші коефіцієнти у розв'язувальному стовпці. Тобто необхідно виконати перетворення, аналогічні тим, що використовуються у алгоритмі Гауса з розв'язання систем лінійних алгебраїчних рівнянь.

1.4. Метод штучного базису

Метод штучного базису – це метод вибору початкового базису. У загальному випадку задача лінійного програмування формулюється з обмеженнями типу $\geq$ і $\leq$. Тому вона буде формуватися не тільки з позитивними додатковими змінними для обмежень типу $\leq$, але і з від'ємними додатковими змінними для обмежень типу $\geq$. Якщо обмеження типу $\leq$, то додаткова змінна x_{j+1} входить у i -те рівняння задачі лінійного програмування із коефіцієнтом, що дорівнює одиниці, і її можна включити до базису. Якщо обмеження типу $\geq$, то додаткова змінна x_{j+1} входить до i -ого рівняння з коефіцієнтом «мінус одиниця», і її включати до базису не можна. Тоді потрібно застосувати такий прийом. У рівняння, які відповідають обмеженням типу $\geq$ чи $=$, вводяться штучні змінні y_i .

Нехай обмеження мають вигляд $\sum_{j=1}^n a_{i,j}x_j = b_i$. ($i=1, 2, \dots, m$), $b_i \geq 0$.

Запишемо їх у вигляді $\sum_{j=1}^n a_{i,j}x_j + y_i = b_i$ ($i=1, 2, \dots, m$), $b_i \geq 0$, де $y_i \geq 0$.

Виберемо y_i як базисну змінну, що відповідає i -му співвідношенню. Змінну y_i називають штучною, тому що вона виникла внаслідок математичного трюку, ціль якого полягає у тому, щоб побудувати початковий розв'язок.

Умова А. На заключній симплекс-ітерації штучна змінна y_i має вийти з базису (перетворитися у нуль). Якщо цього не відбулося і на останній симплекс-ітерації, принаймні, одна з y_i увійде до базису із додатним значенням, то це означає несумісність умов задачі.

Виконання умови А можна забезпечити методом штучного базису. Додамо до цільової функції суму всіх штучних змінних y_i із загальним коефіцієнтом M , значення якого виберемо достатньо великим:

$$x_0 - \sum_{j=1}^n c_j x_j + \sum_{i=1}^m M y_i = 0,$$

Коефіцієнт M має сенс негативного питомого прибутку. Наступним кроком є виключення всіх змінних y_i з цільової функції. Помножимо

рівняння обмежень зі штучними змінними на $-M$. Додамо до цільової функції отримані рівняння. У результаті отримаємо

$$x_0 - \sum_{j=1}^n c_j x_j - \sum_{i=1}^m \sum_{j=1}^n a_{ij} x_j = - \sum_{i=1}^m M y_i .$$

1.5. Графічна інтерпретація симплекс-методу

Розглянемо наступну задачу: фірма виробляє продукцію трьох видів (продукт 1, продукт 2, продукт 3) і закуповує потрібну сировину у двох постачальників. Сировина у постачальників відрізняється за властивостями, а обсяги продуктів 1, 2 і 3, які можна виробити з сировини першого постачальника відрізняються від обсягів продуктів 1, 2 і 3, які можна виробити з сировини другого постачальника. Крім того, є обмеження на обсяги виробництва продуктів 1, 2 і 3. Відповідні показники наведено в табл. 1.3.

Таблиця 1.3

Продукт	Постачальник 1	Постачальник 2	Обмеження на обсяги виробництва
1	0,2	0,3	1,8
2	0,2	0,1	1,2
3	0,3	0,3	2,4
Відносний прибуток	5	6	

Із таблиці випливає, що із 1 т сировини постачальника 1 можна виробити 0,2 т продукту 1, із 0,2 т продукту 2 і 0,3 т продукту 3. Відповідно, 0,3 т складають відходи сировини.

Ціни на сировину у постачальників 1 і 2 можуть відрізнятися. Під «відносним прибутком» розуміють різницю між повною виручкою, що можна отримати з однієї тони сировини, закупленої у відповідного виробника, і вартістю закупівлі і доставки цієї тони сировини.

Змінними x_1 і x_2 позначимо масу сировини, яку потрібно закупити у першого та другого постачальників відповідно. Сформулюємо оптимізаційну задачу

$$L = 5x_1 + 6x_2 \rightarrow \max ;$$

$$0,2x_1 + 0,3x_2 \leq 1,8;$$

$$0,2x_1 + 0,1x_2 \leq 1,2;$$

$$0,3x_1 + 0,3x_2 \leq 2,4;$$

$$x_1 \geq 0, \quad x_2 \geq 0.$$

Умови $x_1 \geq 0$, $x_2 \geq 0$ введені тому, що від'ємні значення цих змінних у цьому випадку не мають фізичного змісту. На рис. 1.1 зображено графіки обмежень та область припустимих стратегій (заштрихована область). Прямі лінії відповідають знаку рівності в обмеженнях і являють собою граничні випадки обмежень. Завдяки знаку нерівності кожне обмеження являє собою напівплощину, яка лежить по один бік прямої. Крім того, на рис. 1.1 показано напрямок градієнта цільової функції L та її лінії рівня (штриховими лініями).

Кожна з паралельних прямих, зображених на рисунку, відповідає різним комбінаціям значень x_1 і x_2 , які призводять до одних і тих самих значень цільової функції $L = 5x_1 + 6x_2$.

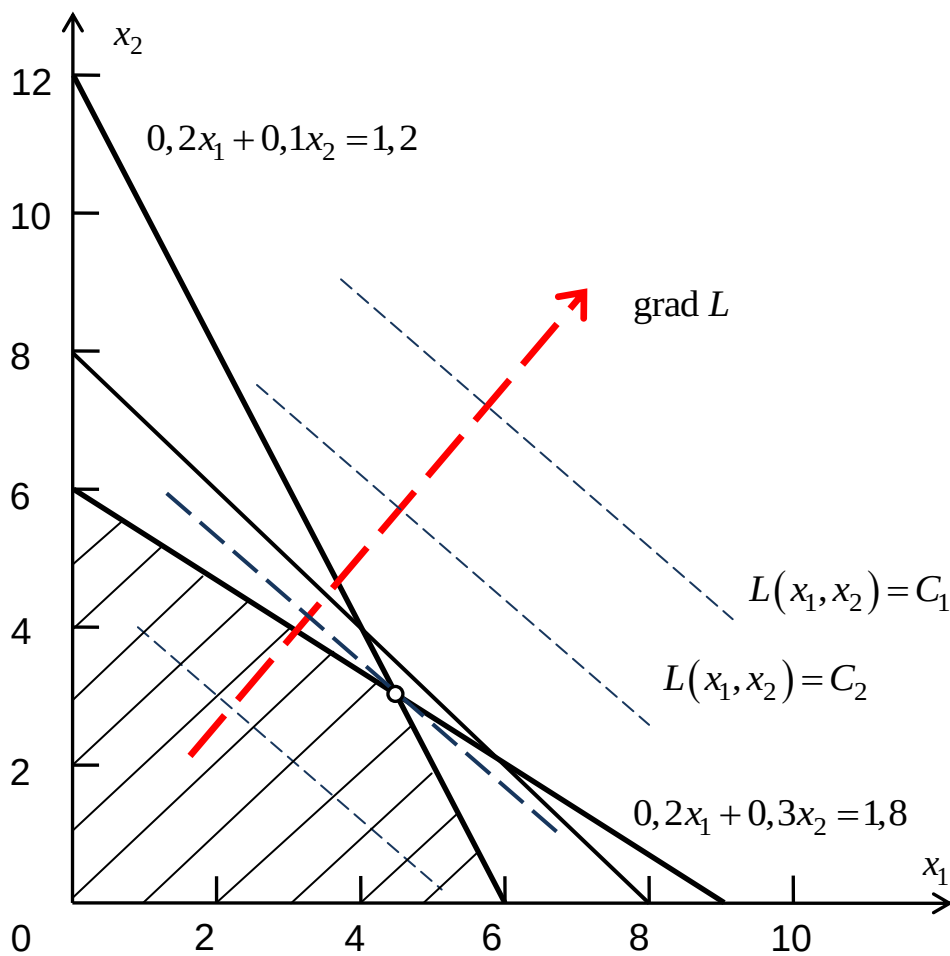


Рис. 1.1. Схема області припустимих значень і градієнт цільової функції

Пряма, яка проходить через точку перехрещення прямих $0,3x_1 + 0,3x_2 = 2,4$ і $0,2x_1 + 0,3x_2 = 1,8$, відповідає оптимальному значенню змінних x_1 і x_2 , які забезпечують максимальне значення цільової функції. Легко переконатися, що цей розв'язок є єдиним. Лінії рівня $L(x_1, x_2) = C_1$ і $L(x_1, x_2) = C_2$, позначені на рис. 1.1, не задовольняють умовам задачі, бо не перетинають область припустимих значень. Розв'язавши систему

$$0,2x_1 + 0,3x_2 = 1,8; \quad 0,2x_1 + 0,1x_2 = 1,2,$$

знаходимо $x_1 = 4,5$ і $x_2 = 3$, тобто оптимальне значення цільової функції дорівнює $L = 40,5$.

Застосуємо для розв'язання цієї задачі симплекс-алгоритм. Задача може бути записана у такому вигляді:

$$\begin{aligned} x_0 - 5x_1 - 6x_2 &= 0 \\ 0,2x_1 + 0,3x_2 + x_3 &= 1,8; \\ 0,2x_1 + 0,1x_2 + x_4 &= 1,2; \\ 0,3x_1 + 0,3x_2 + x_5 &= 2,4. \end{aligned}$$

Розв'язання цієї задачі максимізації наведено в табл. 1.4

Таблиця 1.4

N	Базис	Базисний розв'язок	x_1	x_2	x_3	x_4	x_5
1	x_0	0	-5	-6			
	x_3	1,8	2/10	3/10	1		
	x_4	1,2	2/10	1/10		1	
	x_5	2,4	3/10	3/10			1
2	x_0	36	-1		20		
	x_2	6	2/3	1	10/3		
	x_4	6/10	2/15		-1/3	1	
	x_5	6/10	1/10		-1		1
3	x_0	82/2			35/2	15/2	
	x_2	3		1	5	-5	
	x_1	9/2	1		-5/2	15/2	
	x_5	2/30			-3/4	-3/4	1

Ітерації симплекс-алгоритму відповідають переміщенню вздовж границі опуклої області існування розв'язку від вершини до вершини. Це

показано на рис. 1.3. На першій початковій ітерації $x_1 = x_2 = 0$ ($x_0 = 0$), на другій ітерації $x_1 = 0$, $x_2 = 6$ ($x_0 = 36$) і на третій ітерації $x_1 = 4,5$, $x_2 = 3$ ($x_0 = 40,5$). Тобто кожна ітерація покращує значення цільової функції.

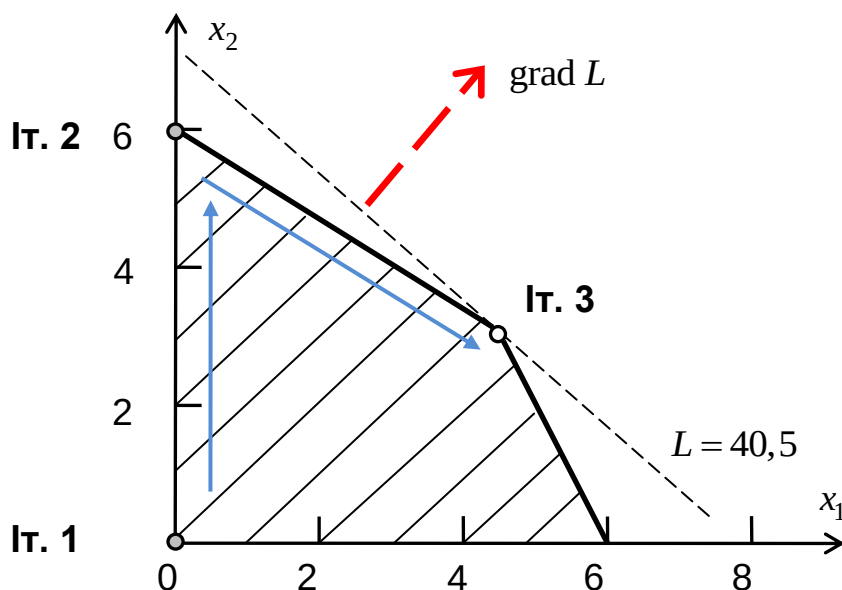


Рис. 1.2. Ітерації симплекс-алгоритму

Двовимірна геометрична інтерпретація дозволяє краще зрозуміти властивості лінійних оптимізаційних моделей і дати інтерпретації понять, які зустрічаються в лінійному програмуванні.

Наявність одного розв'язку. Класична задача має оптимальний розв'язок у точці, яка знаходиться у вершині опуклої області, яку створюють обмеження на змінні, (рис. 1.3.) Лінії рівня цільової функції показано штриховою лінією, градієнт – стрілкою.

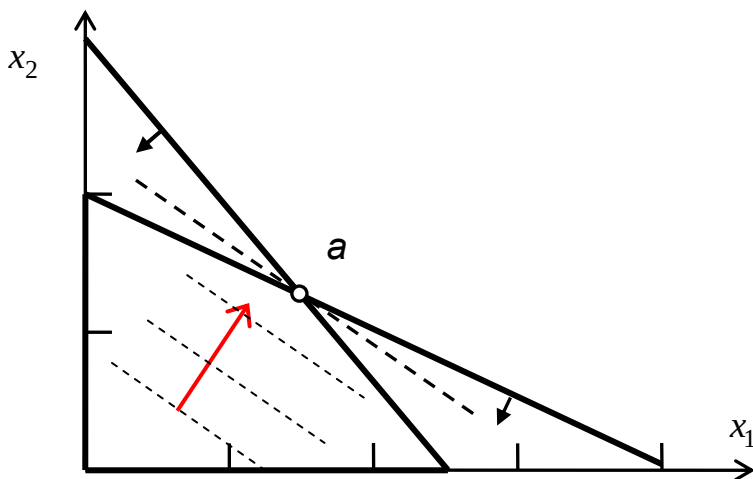


Рис. 1.3. Оптимізаційна модель має єдиний розв'язок (точка а)

Альтернативні оптимальні розв'язки. Градієнт може бути ортогональним до одного з обмежень. У цьому випадку лінії рівня цільової

функції паралельні поверхням (прямим лініям) границі області існування розв'язку. Внаслідок цього всі точки відповідної поверхні (прямої) є оптимальними (рис. 1.4.).

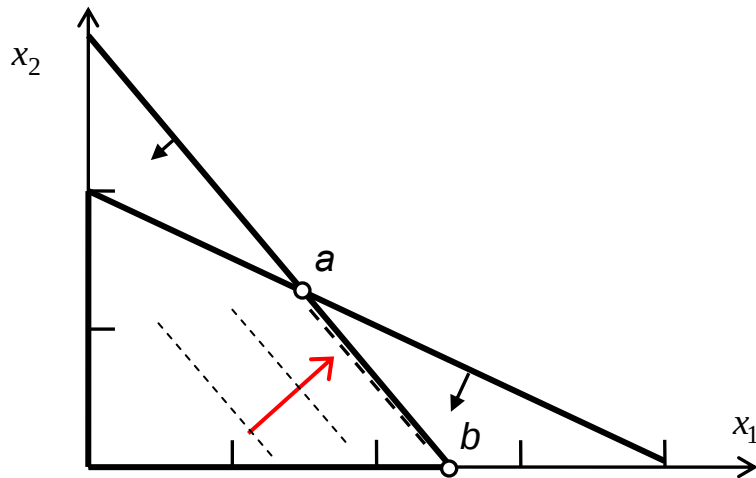


Рис. 1.4. Оптимізаційна модель має альтернативні розв'язки (відрізок ab)

Слід відзначити, що в реальній ситуації, коли кількість змінних у задачі велика (сотні і тисячі змінних), а параметри задачі визначаються за допомогою, наприклад, статистичного аналізу, методів прогнозування і т.п., ситуація, коли градієнт цільової функції ортогональний одному з обмежень, майже неймовірна.

Необмежені оптимальні розв'язки. На рис. 1.5 показано область припустимих значень деякої абстрактної задачі та лінії рівня цільової функції. Ця задача має необмежену множину розв'язків, і значення цільової функції може бути скільки завгодно великим.

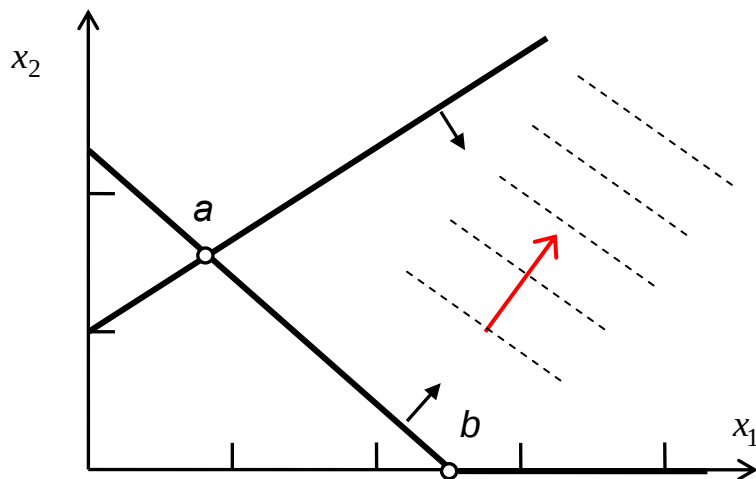


Рис. 1.5. Оптимізаційна модель має необмежений оптимальний розв'язок

Задача, яка не має розв'язку. Обмеження можуть бути такими, що область існування розв'язку не існує (порожня множина). Двовимірний приклад, який ілюструє цю ситуацію, показано на рис. 1.6.

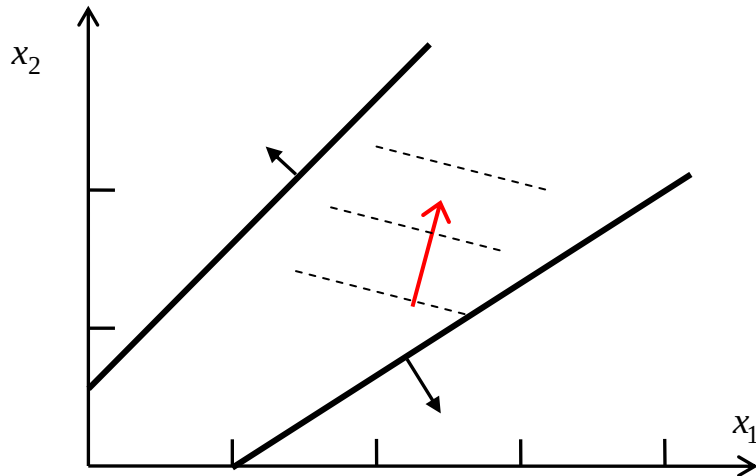


Рис. 1.6. Розв'язку не існує

Розглянуті вище ілюстрації можна пояснити так:

1. Якщо множина розв'язків є не пустою, то вона випукла і може бути або обмеженою, або необмеженою.
2. Якщо множина розв'язків є не пустою, то оптимальний розв'язок може бути або скінченним, або необмежено великим. У випадку, коли оптимальне значення існує, воно відповідає вершині області.

1.6. Аналіз моделей на чутливість

Зазвичай вхідні параметри задачі (такі як, наприклад, продуктивність виробництва, якість сировини, собівартість транспортування, тощо) можуть змінюватися в деяких діапазонах. При розв'язанні економічних задач часто виникає потреба у встановленні діапазонів, у яких можна змінювати вхідні параметри задачі без суттєвого відхилення від знайденого оптимуму і без порушення структури базису, який формує оптимальний розв'язок. Дослідження, яке дозволяє це зробити, має назву «аналіз моделі на чутливість» (або «аналіз моделі при відомому оптимальному розв'язку»).

Проводити аналіз будемо на прикладі, який розглянуто у розд. 1.3, а саме (1.5) та (1.6).

На першому етапі аналізу з'ясуємо, чи залишиться вже знайдений базис оптимальним, якщо змінити коефіцієнти в цільовій функції при небазисних змінних. Наприклад, припустимо, що коефіцієнт при x_2 у (1.5) має невід'ємне прирощення δ , тобто перший рядок системи (1.5) набуває вигляду

$$x_0 - 4x_1 - (5 + \delta)x_2 - 9x_3 - 11x_4 = 0.$$

При виконанні всіх симплекс-ітерацій ми додавали до першого рядка один з інших рядків, попередньо помножив останній на деяку константу. Отже, на заключній ітерації перший рядок системи (1.6) набуде вигляду

$$x_0 + \left(\frac{3}{7} - \delta\right)x_2 + \frac{11}{7}x_4 + \frac{13}{7}x_5 + \frac{5}{7}x_7 = \frac{695}{7}.$$

Таким чином, член $-\delta x_2$ залишиться у першому рядку на будь-якому етапі обчислень.

Ми бачимо, що якщо $x_2 > \frac{3}{7}$, то коефіцієнт при x_2 набуває від'ємного значення. У цьому випадку за симплекс-критерієм I (максимізація) до чергового базису увійшла б змінна x_2 . Аналогічно, якщо коефіцієнт при x_4 набув би значення більше за $\frac{11}{7}$, то базисний розв'язок перестав би бути оптимальним.

Отже, коефіцієнти при небазисних змінних у першому рядку на етапі заключної ітерації показують, у яких границях відповідні коефіцієнти у виразі для цільової функції можуть набувати додатного прирощення без порушення оптимальності раніше отриманого базису.

Припустимо, що змінився коефіцієнт при базисній змінній x_1 . Перший рядок системи (1.5) набуває вигляду

$$x_0 - (4 + \delta)x_1 - 5x_2 - 9x_3 - 11x_4 = 0.$$

А перший рядок у (1.6) набуває відповідно вигляду

$$x_0 - \delta x_1 + \frac{3}{7}x_2 + \frac{11}{7}x_4 + \frac{13}{7}x_5 + \frac{5}{7}x_7 = \frac{695}{7}, \quad (1.7)$$

Щоб дійти до висновку відносно того, в яких межах можна змінювати δ , не порушуючи оптимальності отриманого розв'язку, необхідно, насамперед, обернути до нуля коефіцієнт при x_1 у першому рядку (1.6). Для цього помножимо на δ другий рядок у (1.6) і додамо його до (1.7). Внаслідок цього будемо мати:

$$x_0 + \left(\frac{3}{7} + \frac{5}{7}\delta\right)x_2 + \left(\frac{11}{7} - \frac{5}{7}\delta\right)x_4 + \left(\frac{13}{7} + \frac{10}{7}\delta\right)x_5 + \left(\frac{5}{7} - \frac{1}{7}\delta\right)x_7 = \frac{695}{7} + \frac{50}{7}\delta. \quad (1.8)$$

Звідки випливає, що при

$$-\frac{3}{5} \leq \delta \leq \frac{11}{5} \quad (1.9)$$

отриманий розв'язок залишається оптимальним. Якщо $-\frac{3}{5} > \delta$, коефіцієнт при x_2 набуває від'ємного значення. У випадку $\delta > \frac{11}{5}$ від'ємним стає

коефіцієнт при x_4 . Отже, якщо значення δ виходить за межі вказаного інтервалу, знайдений базис перестане бути оптимальним.

Описаний вище прийом придатний і у випадку одночасного змінювання значень коефіцієнтів при декількох базисних змінних. При цьому в співвідношення, яке є аналогом співвідношення (1.8), увійдуть декілька змінних прирощень δ_j , кожне з яких впливає на спосіб вибору нового базису. Щоб базис залишився незмінним, δ_j повинні підпорядковуватися деяким умовам, записаним у вигляді нерівностей, аналогічних (1.9). Тобто в цьому випадку будемо мати замість однієї нерівності (1.9) систему нерівностей, кожне з яких отримується з умови невід'ємності коефіцієнтів при небазисних змінних.

Відомі комп'ютерні програми, побудовані на базі симплекс-методу, автоматично забезпечують класифікаційний аналіз (ранжування) коефіцієнтів у виразі для цільової функції. Існують стандартні програми, які передбачають більш тонкий аналіз лінійних моделей на чутливість, при цьому застосовуються такі обчислювальні прийоми, як метод приєднаних цільових функцій і параметричне програмування.

Окрім варіювання коефіцієнтів при змінних у цільової функції для системного аналітика є інтерес дослідження впливу на розв'язок варіювання коефіцієнтів у правих частинах обмежень. Наприклад, число 15 у правій частині другого рядка системи (1.5) замінимо на $15+\delta$ і будемо при виконанні ітерацій симплекс-методу враховувати цю заміну. На останній ітерації отримаємо:

$$\begin{aligned}x_0 &+ \frac{3}{7}x_2 + \frac{11}{7}x_4 + \frac{13}{7}x_5 + \frac{5}{7}x_7 = \frac{695}{7} + \frac{13}{7}\delta, \\x_1 &+ \frac{5}{7}x_2 - \frac{5}{7}x_4 + \frac{10}{7}x_5 - \frac{1}{7}x_7 = \frac{50}{7} + \frac{10}{7}\delta, \\&-\frac{6}{7}x_2 + \frac{13}{7}x_4 - \frac{61}{7}x_5 + x_6 + \frac{4}{7}x_7 = \frac{325}{7} - \frac{61}{7}\delta, \\&\frac{2}{7}x_2 + x_3 + \frac{12}{7}x_4 - \frac{3}{7}x_5 + \frac{1}{7}x_7 = \frac{55}{7} - \frac{3}{7}\delta.\end{aligned}$$

Покладаючи значення небазисних змінних такими, що дорівнюють нулеві, маємо значення базисних змінних, які визначаються через δ . Щоб базис залишався допустимим, константи у правих частинах повинні мати невід'ємні значення. Звідси слідує, що отриманий розв'язок залишиться допустимим, якщо

$$-\frac{50}{10} \leq \delta \leq \frac{325}{61}.$$

Дійсно, при $\delta < -\frac{50}{10}$ значення базисної змінної x_1 стає від'ємним; при $\delta > \frac{325}{61}$ від'ємного значення набуває базисна змінна x_6 .

Розглянутий метод аналізу можна узагальнити на випадок варіювання одночасно декількох констант у правих частинах обмежень, які входять до складу системи. Виконаємо в правих частинах (1.5) такі заміни:

$$15 \rightarrow (15 + \delta_1),$$

$$120 \rightarrow (120 + \delta_2),$$

$$100 \rightarrow (100 + \delta_3).$$

Після виконання всіх операцій, які призводять систему (1.5) до вигляду (1.6) і перетворення до нуля всіх небазисних змінних, будемо мати:

$$L = x_0 = \frac{695}{7} + \frac{13}{7}\delta_1 + \frac{5}{7}\delta_3,$$

$$x_1 = \frac{50}{7} + \frac{10}{7}\delta_1 - \frac{1}{7}\delta_3,$$

$$x_6 = \frac{325}{7} - \frac{61}{7}\delta_1 + \delta_2 + \frac{4}{7}\delta_3,$$

$$x_3 = \frac{55}{7} - \frac{3}{7}\delta_1 + \frac{1}{7}\delta_3.$$

Коефіцієнти при δ_1 , δ_2 , δ_3 збігаються з коефіцієнтами при відповідних вільних змінних у (1.6). Базис залишається припустимим, якщо значення x_1 , x_6 , x_3 невід'ємні. Тобто, δ_1 , δ_2 , δ_3 повинні задовольняти відповідну систему нерівностей.

1.7. Двоїстість

Кожній задачі лінійного програмування із змішаними обмеженнями, що мають вид

$$L = \sum_{j=1}^n c_j x_j \rightarrow \max, \quad (1.10)$$

$$\sum_{j=1}^n a_{i,j} x_j \leq b_i, \quad i = 1, 2, \dots, m,$$

$$x_j \geq 0, \quad j = 1, 2, \dots, n,$$

можна поставити у відповідність іншу задачу, яка називається двоїстою по відношенню до першої:

$$Z = \sum_{i=1}^m b_i y_i \rightarrow \min, \quad (1.11)$$

$$\sum_{i=1}^m a_{i,j} y_i \geq c_j, \quad j=1, 2, \dots, m,$$

$$y_j \geq 0, \quad j=1, 2, \dots, m.$$

Для визначеності умовно назвемо задачу (1.10) вихідною, а другу (1.11) – двоїстою (по відношенню до першої задачі (1.10)). Пряма і двоїста задачі називаються спряженими.

Зрівнюючи форми запису прямої і двоїстої задач, можна встановити такі взаємозв'язки:

1) якщо пряма задача є задачею максимізації, то двоїста буде задачею мінімізації, і навпаки;

2) коефіцієнти цільової функції прямої задачі $c_1, c_2, \dots, c_n$ стають вільними членами обмежень двоїстої задачі;

3) вільні члени обмежень прямої задачі $b_1, b_2, \dots, b_m$ стають коефіцієнтами цільової функції двоїстої задачі;

4) матрицю обмежень двоїстої задачі знаходять транспонуванням матриці обмежень прямої задачі;

5) число змінних двоїстої задачі дорівнює числу обмежень прямої задачі, а число обмежень двоїстої дорівнює числу змінних прямої задачі, і навпаки;

6) взаємно однозначна відповідність між змінними прямої задачі і обмеженнями двоїстої задачі задовольняє наступному положенню: j -е обмеження двоїстої задачі буде нерівністю, якщо на j -ю змінну початкової задачі накладена вимога позитивності, в протилежному випадку j -е обмеження буде рівністю.

Теорема 1.1 (основна (перша) теорема двоїстості). Якщо вихідна задача і двоїста до неї мають припустимі розв'язки, то: існує оптимальний розв'язок $x_j^* (j=1, 2, \dots, n)$ вихідної задачі: існує оптимальний розв'язок $y_i^* (i=1, 2, \dots, m)$ двоїстої задачі; має місце відношення

$$\max L = \min Z,$$

тобто

$$\sum_{j=1}^n c_j x_j^* = \sum_{i=1}^m b_i y_i^*.$$

Якщо вихідна (двоїста) задача припускає оптимальний розв'язок, для якого значення цільової функції обмежено, то відповідна їй двоїста (вихідна) задача припускає оптимальний розв'язок при цьому ж значенні цільової функції.

Теорема про додаткову нежорсткість (друга теорема двоїстості).

Нехай $x_j^* (j=1, 2, \dots, n)$ - розв'язок вихідної задачі, а $y_i^* (i=1, 2, \dots, m)$ - розв'язок відповідної двоїстої задачі. Обидва розв'язки є оптимальними тоді і тільки тоді, коли

$$y_i^* \left(\sum_{j=1}^n a_{i,j} x_j^* - b_i \right) = 0, \quad i=1, 2, \dots, m;$$
$$x_j^* \left(\sum_{i=1}^m a_{i,j} y_i^* - c_j \right) = 0, \quad j=1, 2, \dots, n.$$

Друга теорема двоїстості стверджує, що якщо хоча б один з оптимальних розв'язків двох спряжених задач перетворює i -те обмеження в строгу нерівність, то відповідна змінна для іншої задачі x_j^* (або y_i^*) дорівнює нулю. Якщо ж i -та координата хоча б одного оптимального розв'язку додатна, то кожний оптимальний розв'язок іншої спряженої задачі перетворює i -те обмеження у рівність.

Економічна інтерпретація двоїстої задачі. Розглянемо наступну задачу: нехай підприємство має m видів ресурсів, запаси яких становлять $b_1, b_2, \dots, b_m$. Ресурси використовуються для виготовлення n видів продукції. Відома ціна кожного виду продукції – $c_1, c_2, \dots, c_n$ і норма витрат $a_{i,j}$ i -го ресурсу на виготовлення одиниці продукції виду j . Потрібно визначити план виробництва кожного виду продукції, який максимізує прибуток від продажу продукції.

Якщо $x_1, x_2, \dots, x_n$ – обсяги виробництва кожного виду продукції, то можна сформулювати задачу лінійного програмування (1.10).

Припустимо тепер, що підприємство планує продати всі наявні ресурси. Необхідно визначити оптимальні ціни на кожний вид ресурсів y_j ($j=1, 2, \dots, m$), тобто такі, щоб втрати при продажу ресурсів були мінімальними. При цьому необхідно врахувати, що загальна виручка має бути не менша, ніж та, яку можна було б отримати, використовуючи ці ресурси для виготовлення продукції. Математична модель поставленої задачі має вигляд (1.11).

Отримана задача є двоїстою до цієї задачі. Отже, можна стверджувати, що, якщо існує оптимальний план випуску продукції при заданих ресурсах, то існує оптимальна вартість ресурсів, для якої загальна вартість усіх ресурсів буде найменшою. Змінні двоїстої задачі виражають цінність одиниці кожного виду ресурсів, тому їх іноді називають тіншовими цінами.

Приклад 1.2. Для задачі

$$\begin{aligned} Z &= 4x_1 - x_2 + 5x_3 - 7x_4 + 8x_5 - x_6 \rightarrow \max \\ -x_1 - x_2 + 12x_3 - 14x_4 + 7x_5 - x_6 &\geq -1, \\ 3x_1 - 6x_2 - 10x_3 - 5x_4 - 7x_6 &\leq 5, \\ 11x_1 + 5x_2 - x_3 + 2x_4 - x_5 + 3x_6 &= 9, \\ x_2 &\geq 0, \quad x_4 \geq 0 \end{aligned}$$

побудувати двоїсту задачу.

Розв'язання. Змінімо знак нерівності у першій нерівності:

$$\begin{aligned} Z &= 4x_1 - x_2 + 5x_3 - 7x_4 + 8x_5 - x_6 \rightarrow \max \\ x_1 + x_2 - 12x_3 + 14x_4 - 7x_5 + x_6 &\leq 1, \\ 3x_1 - 6x_2 - 10x_3 - 5x_4 - 7x_6 &\leq 5, \\ 11x_1 + 5x_2 - x_3 + 2x_4 - x_5 + 3x_6 &= 9, \\ x_2 &\geq 0, \quad x_4 \geq 0. \end{aligned}$$

Двоїстою до цієї задачі буде задача

$$\begin{aligned} Y &= 1y_1 + 5y_2 + 9y_3 \rightarrow \min \\ \begin{cases} y_1 + 3y_2 + 11y_3 = 4, \\ y_1 - 6y_2 + 5y_3 \geq -1, \\ -12y_1 - 10y_2 - y_3 = 5, \\ 14y_1 - 5y_2 + 2y_3 \geq -7, \\ -7y_1 - y_3 = 8, \\ y_1 - 7y_2 + 3y_3 = -1. \end{cases} \\ y_1 &\geq 0, \quad y_2 \geq 0 \end{aligned}$$

Приклад. Для задачі

$$\begin{aligned} Z &= 2x_1 + 10x_2 + 4x_3 + 2x_4 \rightarrow \min \\ \begin{cases} x_1 + 2x_2 + 2x_3 - 2x_4 = 2, \\ -x_1 + x_2 + x_3 + x_4 = 3, \end{cases} \end{aligned}$$

де $x_i \geq 0, i = 1, 2, 3, 4$;

побудувати двоїсту задачу і знайти її розв'язок графічним методом.

Розв'язання. Перейдемо до задачі максимізації:

$$\begin{aligned} L &= -2x_1 - 10x_2 - 4x_3 - 2x_4 \rightarrow \max, \\ \begin{cases} x_1 + 2x_2 + 2x_3 - 2x_4 = 2, \\ -x_1 + x_2 + x_3 + x_4 = 3, \\ x_i \geq 0, i = 1, 2, 3, 4. \end{cases} \end{aligned}$$

Двоїстою до цієї задачі буде задача

$$Y = 2y_1 + 3y_2 \rightarrow \min$$

$$\begin{cases} y_1 - y_2 \geq -2, \\ 2y_1 + y_2 \geq -10, \\ 2y_1 + y_2 \geq -4, \\ -2y_1 + y_2 \geq -2. \end{cases}$$

Обмеження на знак нових змінних y_1, y_2 не накладаються. Побудуємо графіки обмежень і лінії рівня цільової функції (рис. 1.7).

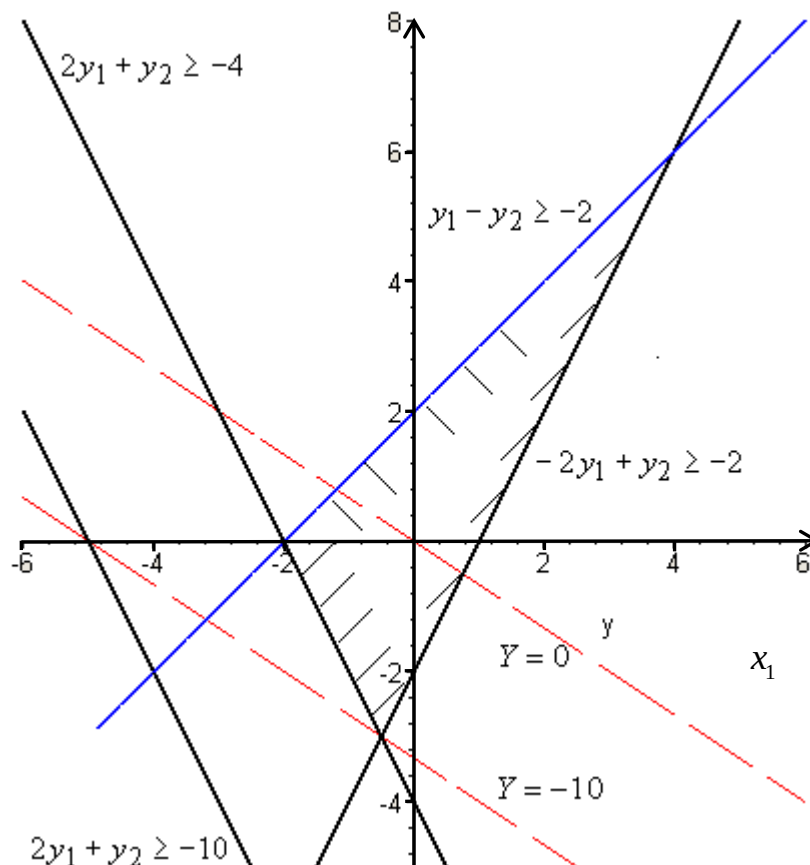


Рис. 1.7. Графічне розв'язання задачі

З рисунка видно, що мінімального значення ($Y = -10$) цільова функція двоїстої задачі досягає в точці перетинання третьої і четвертої нерівностей. Відповідно в цій точці обмеження виконуються як рівності, а перше і друге обмеження – як строгі нерівності. Змінні x_1 і x_2 , які відповідають цим нерівностям, дорівнюють нулю. Маємо:

$$\begin{cases} 2x_3 - 2x_4 = 2, \\ x_3 + x_4 = 3. \end{cases}$$

Звідки знаходимо $x_3 = 2, x_4 = 1$. Відповідно цільова функція вихідної задачі $L = -2 \cdot 0 - 10 \cdot 0 - 4 \cdot 2 - 2 \cdot 1 = -10$, що збігається зі значенням, отриманим вище за допомогою переходу до двоїстої задачі.

1.8. Двоїстий симплекс-алгоритм

Спираючись на зв'язок між вихідною задачею і двоїстою до неї, можна побудувати ще один метод розв'язання задачі лінійного програмування. Розглянемо такий алгоритм розв'язання вихідної задачі, для якого на кожній ітерації, за винятком останньої, розв'язок є неприпустимим внаслідок невиконання умов невід'ємності змінних. Відповідний же розв'язок двоїстої задачі на кожній ітерації є припустимим. Ця ідея лежить в основі так званого двоїстого симплекс-алгоритму.

Почнемо ознайомлення з дією двоїстого симплекс-алгоритму, користуючись таким прикладом:

Приклад 1.3.

$$\begin{aligned} L &= 2x_1 + x_3 \rightarrow \min, \\ x_1 + x_2 - x_3 &\geq 5, \\ x_1 - 2x_2 + 4x_3 &\geq 8, \\ x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0. \end{aligned} \tag{1.12}$$

Відповідна до неї двоїста задача має вигляд:

$$\begin{aligned} Z &= 5y_1 + 8y_2 \rightarrow \max \\ y_1 + y_2 &\leq 2, \\ y_1 - 2y_2 &\leq 0, \\ -y_1 + 4y_2 &\leq 1, \\ y_1 \geq 0, \quad y_2 \geq 0. \end{aligned}$$

Якщо позначити x_0 значення цільової функції і ввести до (1.12) додаткові змінні, то вихідна задача прийме вигляд:

$$\begin{aligned} x_0 - 2x_1 - x_3 &= 0, \\ x_1 + x_2 - x_3 - x_4 &= 5, \\ x_1 - 2x_2 + 4x_3 - x_5 &= 8. \end{aligned} \tag{1.13}$$

Якщо включити в перший пробний розв'язок змінні x_0, x_4, x_5 як базисні, то цей розв'язок буде неприпустимим, оскільки змінні x_4, x_5 приймуть від'ємні значення, а x_0 буде дорівнювати нулю. Однак цей базис є припустимим для двоїстої задачі. Метод, що пропонується, полягає в тому, що на кожній ітерації (окрім останньої) знаходиться припустимий розв'язок двоїстої задачі.

Насамперед, сформулюємо правило, за яким визначаємо змінну, яку потрібно **виключити** з пробного базису.

Двоїстий симплекс-критерій I. Якщо базис містить змінні, які приймають від'ємні значення, то виключенню з базису підлягає одна з цих змінних, а саме та змінна, значення якої є максимальним за модулем. Якщо всі базисні змінні мають невід'ємні значення, то такий розв'язок є оптимальним.

Для зручності запишемо (1.13) у вигляді

$$\begin{aligned}x_0 - 2x_1 - x_3 &= 0, \\ -x_1 - x_2 + x_3 + x_4 &= -5, \\ -x_1 + 2x_2 - 4x_3 + x_5 &= -8.\end{aligned}$$

Згідно з симплекс-критерієм I, виключенню з базису підлягає x_5 . Сформулюємо тепер правило вибору змінної, яка підлягає **включенню** до базису.

Двоїстий симплекс-критерій II (мінімізація). 1) Беремо відношення значень коефіцієнтів при небазисних змінних у першому рядку до відповідних коефіцієнтів, що фігурують в рядку, який містить змінну, яка виключається з базису. При цьому не враховуються відношення, знаменники в яких дорівнюють нулю, або додатному числу. 2) Вибираємо мінімальне відношення, яке відповідає деякій змінній. Ця змінна має бути включена до нового базису.

Результати обчислень наведено у табл. 1.5.

Таблиця 1.5

Небазисні змінні	x_1	x_2	x_3
Коефіцієнти в рядку 1	-2	0	-1
Коефіцієнти в рядку 3	-1	2	-4
Відношення	2	-	0,25

З таблиці видно, що включенню до базису підлягає x_3 . Отримаємо:

$$\begin{aligned}x_0 - \frac{7}{4}x_1 - \frac{2}{4}x_2 - \frac{1}{4}x_5 &= 2, \\ -\frac{5}{4}x_1 - \frac{2}{4}x_2 + x_4 + \frac{1}{4}x_5 &= -7, \\ \frac{1}{4}x_1 + \frac{2}{4}x_2 + x_3 - \frac{1}{4}x_5 &= 2.\end{aligned}$$

Ітерація 2. Повертаючись до критерію I, приходимо до висновку, що з базису потрібно виключити змінну x_4 . У табл. 1.6 наведено результати розрахунків, виконаних згідно з вимогами критерію II. Ми бачимо, що в черговий базис замість x_4 увійде змінна x_2 .

Таблиця 1.6

Небазисні змінні	x_1	x_2	x_5
Коефіцієнти в рядку 1	-7/4	-2/4	-1/4
Коефіцієнти в рядку 2	-5/4	-2/4	1/4
Відношення	7/5	1	-

Після виконання всіх перетворень, пов'язаних з включенням до базису x_2 , отримаємо

$$\begin{aligned} x_0 - \frac{1}{2}x_1 - x_4 - \frac{1}{4}x_5 &= 9, \\ \frac{5}{2}x_1 + x_2 - 2x_4 - \frac{1}{2}x_5 &= 14, \\ \frac{3}{2}x_1 + x_3 - x_4 - \frac{1}{2}x_5 &= 9. \end{aligned}$$

Значення всіх базисних змінних є невід'ємними, і, відповідно, оптимальний розв'язок отримано. При цьому $x_0 = 9$, $x_2 = 14$, $x_3 = 9$.

Розглянутий вище алгоритм легко перетворити для розв'язання задач максимізації. Двоїстий симплекс-критерій I залишається без змін, а формулювання симплекс-критерію II необхідно змінити потрібним чином.

Двоїстий симплекс-критерій II (максимізація). 1. Беремо відношення значень коефіцієнтів при небазисних змінних у першому рядку до відповідних коефіцієнтів, що фігурують у рядку, який містить змінну, яка виключається з базису. При цьому не враховуються відношення, знаменники в яких дорівнюють нулю, або додатному числу. 2. Вибираємо максимальне відношення, яке відповідає деякій змінній. Ця змінна має бути включена до чергового базису.

1.9. Збіжність симплекс-алгоритму

Симплекс-метод складається з переходів від одного базисного розв'язку до іншого. Оскільки значення змінних для кожного пробного розв'язку визначені однозначно, припущення про обов'язкове покращення цільової функції при переході від одного до іншого базису виключає повторення у виборі пробних базисів. Кількість пробних базисів обмежена, і для n -розмірної задачі з m обмеженнями не може перевищувати кількість комбінацій із n по m , тобто $C_n^m = \frac{n!}{m!(n-m)!}$. Відповідно кількість

ітерації обмежена, і розв'язок, отриманий на останній ітерації, є оптимальним.

Чи можна вважати, що збіжність завжди має місце? На жаль, ні. Для багатьох лінійних оптимізаційних моделей покращення цільової функції на деяких ітераціях призупиняється. Це відбувається у такій послідовності.

На одній з ітерацій за критерієм II може з'ясуватися, що серед усіх відношень правих частин рівнянь, які задають обмеження, до коефіцієнтів при новій базисній змінній найменше відношення дорівнює нулю. Така ситуація може виникнути у тому випадку, коли в результаті включення до базису нової змінної на попередній ітерації дві або більше двох змінних, які раніше входили до базису, обертаються в нуль. Таким чином, при ітерації що розглядається, значення по найменш однієї базисної змінної опиняється таким, що дорівнює нулю, тобто базисний розв'язок є виродженим. Звідки слідує, що нова базисна змінна приймає нульове значення, а значення інших змінних, а також і значення цільової функції, не змінюються. Зміна базису призводить до зміни коефіцієнтів у першому рядку. Тому на етапі застосування критерію II на наступній ітерації найменше зі всіх відношень вказаного вище типу виявляється строго додатним. Однак можлива нова затримка у покращенні цільової функції. Такі ситуації можуть виникати неодноразово, причому не виключено, що на якому-небудь етапі ми прийдемо до одного з попередніх базисних розв'язків. У цьому випадку обчислювальний процес виявляється «зацикленим» і збіжність відсутня. Але така ситуація виникає рідко, і ймовірність її виникнення в реальній задачі, де коефіцієнти рідко бувають цілими або кратними числами, близька до нуля.

Досвід показує, що кількість ітерацій, яка потрібна для розв'язання більшості практичних задач, лежить у межах від $1,5m$ до $3m$, де m - кількість обмежень. Але сумарна кількість операцій у грубому наближенні пропорційна кубу кількості обмежень.

Крім того, для задач великої розмірності (тисячі обмежень і більше) проблемою становить накопичення похибок при округленні коефіцієнтів при розрахунках на кожній ітерації. Ця проблема загострюється, якщо коефіцієнти при змінних відрізняються на декілька порядків, тому що при виключенні змінних з базису виникає потреба ділити ці коефіцієнти, що призводить до округлення і втрати точності. Тому рекомендується перевіряти точність розв'язку шляхом підстановки отриманих результатів у початкові рівняння, які задають обмеження.

2. ЦІЛОЧИСЛОВЕ ПРОГРАМУВАННЯ

2.1. Задачі цілочислового програмування

Екстремальна задача лінійного програмування, змінні якої набувають тільки цілочислових значень, називається задачею цілочислового лінійного програмування.

Приклад 2.1. У цеху підприємства вирішено встановити додаткове устаткування, для розміщення якого виділено $\frac{19}{3} \text{ м}^2$ площі. На придбання устаткування підприємство може витратити 10 тис. грн, при цьому воно може купити устаткування двох видів. Комплект устаткування 1-го виду коштує 1000 грн, а 2-го виду – 3000 грн. Придбання одного комплекту устаткування 1-го виду дозволяє збільшити випуск продукції за зміну на 2 од., а одного комплекту устаткування 2-го виду — на 4 од. Знаючи, що для встановлення одного комплекту устаткування 1-го виду потрібно 2 м^2 площі, а устаткування 2-го виду — 1 м^2 площі, визначити такий набір додаткового устаткування, який дає можливість максимально збільшити випуск продукції.

Розв'язання. Складемо математичну модель задачі. Припустимо, що підприємство придбає x_1 комплектів устаткування 1-го виду і x_2 комплектів устаткування 2-го виду. Тоді змінні x_1 і x_2 повинні задовольняти таким нерівностям:

$$\begin{cases} 2x_1 + x_2 \leq 19/3, \\ x_1 + 3x_2 \leq 10. \end{cases} \quad (2.1)$$

Якщо підприємство придбає зазначену кількість устаткування, то загальне збільшення випуску продукції складе

$$F = 2x_1 + 4x_2. \quad (2.2)$$

За своїм економічним змістом змінні x_1 і x_2 можуть набувати тільки цілих невід'ємних значень, тобто

$$x_1, x_2 \geq 0, \quad (2.3)$$

$$x_1, x_2 - \text{цілі}. \quad (2.4)$$

Таким чином, приходимо до такої математичної задачі: знайти максимальне значення лінійної функції (2.2) при виконанні умов (2.1), (2.3) і (2.4). Оскільки невідомі можуть мати тільки цілі значення, то задача (2.1) – (2.4) є задачею цілочислового програмування. Оскільки число невідомих задачі дорівнює двом, розв'язок цієї задачі можна знайти, використовуючи її геометричну інтерпретацію. Для цього, перш за все, побудуємо багатокутник розв'язків задачі, що полягає у визначенні максимального значення лінійної функції (2.2) при виконанні умов (2.1) і (2.3) (рис. 2.1).

Координати всіх точок побудованого багатокутника розв'язків ОАЕВС задовольняють системі лінійних нерівностей (2.1) і умові невід'ємності змінних (2.3). Разом з тим умові (2.4), тобто умові цілочисловості змінних задовольняють координати лише 12 точок, позначених на рис. 2.1.

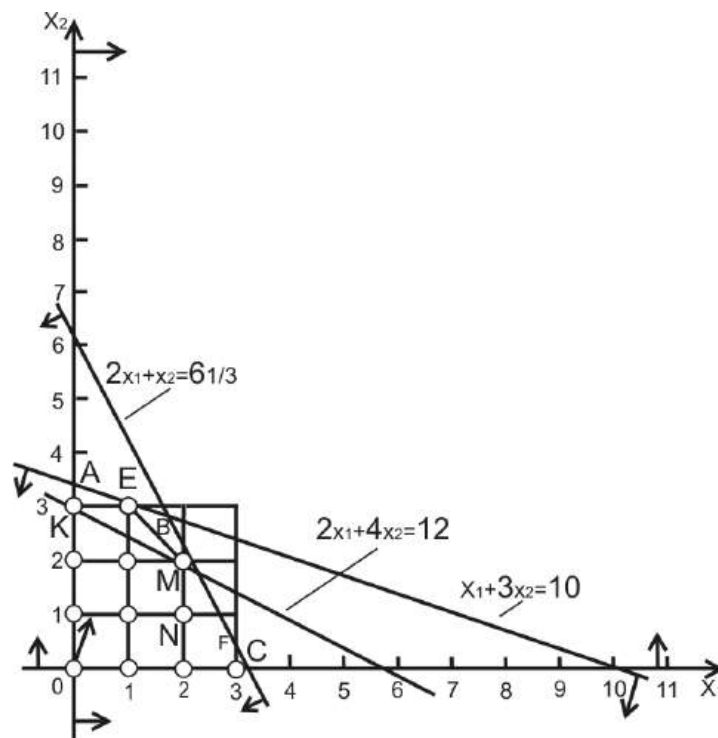


Рис. 2.1. Схема області припустимих значень і градієнт цільової функції

Щоб знайти точку, координати якої визначають розв'язок початкової задачі, замінимо багатокутник ОАВС багатокутником ОКЕМНF, що містить усі допустимі точки з цілочисловими координатами і таким, що координати кожної з вершин є цілими числами. Значить, якщо знайти точку максимуму функції (2.2) на багатокутнику ОКЕМНF, то координати цієї точки і визначають оптимальний план задачі.

Для цього побудуємо вектор $\vec{C} = (2; 4)$ і пряму $2x_1 + 4x_2 = 12$, що проходить через багатокутник розв'язків ОКЕМНF (число 12 узято довільно). Побудовану пряму рухаємо у напрямі вектора $\vec{C}$ доти, поки вона не пройде через останню спільну точку її з даним багатокутником. Координати цієї точки і визначають оптимальний план, а значення цільової функції в ній є максимальним.

У наведеному випадку шуканою є точка $E = (1; 3)$, в якій цільова функція набуває максимального значення $F_{\max} = 14$. Отже, координати точки E визначають оптимальний план задачі (2.1) – (2.4). Відповідно до цього плану підприємству слід придбати один комплект устаткування 1-го виду і три комплекти устаткування 2-го виду. Це забезпечить підприємству при наявних у нього обмеженнях на виробничі площі й грошові кошти максимальне збільшення випуску продукції, що дорівнює 14 од. за зміну.

2.2. Метод Гоморі

Розглянемо задачу:

$$F = \sum_{j=1}^n c_j x_j \rightarrow \max \quad (2.5)$$

$$\sum_{j=1}^n a_{ij} x_j = b_i \quad (i = \overline{1, m}), \quad (2.6)$$

$$x_j \geq 0 \quad (j = \overline{1, n}), \quad (2.7)$$

$$x_j \text{ — цілі } (j = \overline{1, n}). \quad (2.8)$$

Розглянемо деякі методи відшукування оптимального плану задачі (2.5) – (2.8).

Знаходження розв'язку задачі цілочислового програмування методом Гоморі починають із визначення симплексним методом оптимального плану задачі (2.5) – (2.7) без врахування умови (2.8). Після того як цей план знайдено, переглядають його компоненти. Якщо серед компонентів немає дробових чисел, то знайдений план є оптимальним планом задачі цілочислового програмування (2.5) – (2.8). Якщо ж в оптимальному плані задачі (2.5) – (2.7) змінна x_j набуває дробового значення, то до системи рівнянь додають нерівність

$$\sum_j f(a_{ij}^*) x_j \geq f(b_i^*) \quad (2.9)$$

і знаходять розв'язок задачі (2.5) – (2.7), (2.9).

У нерівності (2.9) a_{ij}^* і b_i^* – перетворені вихідні величини a_{ij} та b_i , значення яких узяті з останньої симплекс-таблиці, а $f(a_{ij}^*)$ і $f(b_i^*)$ – дробові частини чисел (під дробовою частиною деякого числа a розуміється найменше невід'ємне число b , таке, що різниця між a і b є ціле число).

Якщо в оптимальному плані задачі (2.5) – (2.7) дробові значення приймають декілька змінних, то додаткова нерівність (2.9) визначається найбільшою дробовою частиною.

Якщо у знайденому плані задачі (2.5) – (2.7), (2.9) змінні набувають дробових значень, то знову додають одне додаткове обмеження й процес обчислень повторюють. Проводячи скінченне число ітерацій, або одержують оптимальний план задачі цілочислового програмування (2.5) – (2.8), або встановлюють її нерозв'язність.

Якщо вимога цілочисловості (2.8) накладається тільки на деякі змінні, то такі задачі називаються частково цілочисловими. Їх розв'язок також знаходять послідовним розв'язуванням задач, кожену з яких отримують із попередньої за допомогою введення додаткового обмеження. У цьому випадку таке обмеження має вигляд

$$\sum_j \gamma_{ij} x_j \geq f(b_i^*), \quad (2.10)$$

де γ_{ij} визначаються з таких співвідношень:

1) для x_j , які можуть набувати нецілочислових значень:

$$\gamma_{ij} = \begin{cases} a_{ij}^*, & \text{при } a_{ij}^* \geq 0, \\ \frac{f(b_i^*)}{1 - f(b_i^*)} |a_{ij}^*|, & \text{при } a_{ij}^* < 0; \end{cases} \quad (2.11)$$

2) для x_j , які можуть набувати тільки цілочислових значень:

$$\gamma_{ij} = \begin{cases} f(a_{ij}^*), & \text{при } f(a_{ij}^*) \leq f(b_i^*), \\ \frac{f(b_i^*)}{1 - f(b_i^*)} |1 - f(a_{ij}^*)|, & \text{при } f(a_{ij}^*) > f(b_i^*). \end{cases} \quad (2.12)$$

Отже, можемо записати алгоритм знаходження оптимального плану цілочислової задачі лінійного програмування методом Гоморі.

1. Використовуючи симплекс-метод, знаходять розв'язок задачі (2.5) – (2.7) без врахування вимоги цілочисловості змінних (2.8).
2. Складають додаткове обмеження для змінної, яка в оптимальному плані задачі (2.5) – (2.7) має максимальне дробове значення, а в оптимальному плані задачі (2.5) – (2.8) повинна бути цілочисловою.
3. Використовуючи двоїстий симплекс-метод, знаходять розв'язок задачі (2.5) – (2.7), (2.9).
4. Якщо є потреба, складають ще одне додаткове обмеження й продовжують ітераційний процес до одержання оптимального плану задачі (2.5) – (2.8) або встановлюють її нерозв'язність.

Приклад 2.2. Методом Гоморі знайти розв'язок задачі

$$F = x_1 - x_2 \rightarrow \max \quad (2.13)$$

за умов

$$\begin{cases} x_1 - 2x_2 + x_3 = 1, \\ x_1 + 3x_2 + x_4 = 3, \end{cases} \quad (2.14)$$

$$x_j \geq 0 \quad (j = \overline{1,4}), \quad (2.15)$$

$$x_j - \text{цілі} \quad (j = \overline{1,4}). \quad (2.16)$$

Розв'язання. Для визначення оптимального плану задачі (2.13) – (2.16) спочатку знаходимо оптимальний план задачі (2.13) – (2.15):

Таблиця 2.1

i	Б	C_σ	P_0	1	-1	0	0
				P_1	P_2	P_3	P_4
1	P_3	0	1	1	-2	1	0
2	P_4	0	3	1	3	0	1
3			0	-1	1	0	0
1	P_1	1	1	1	-2	1	0
2	P_4	0	2	0	5	-1	10
3			1	0	-1	1	0
1	P_1	1	$\frac{9}{5}$	1	0	$\frac{3}{5}$	$\frac{2}{5}$
2	P_2	-1	$\frac{2}{5}$	0	1	$-\frac{1}{5}$	$\frac{1}{5}$
3			$\frac{7}{5}$	0	0	$\frac{4}{5}$	$\frac{1}{5}$

Як видно з таблиці 2.1, знайдений оптимальний план $X = \left(\frac{9}{2}; \frac{2}{5}; 0; 0\right)$ задачі (2.13) – (2.15) не є оптимальним планом задачі (2.13) – (2.16), оскільки дві змінні x_1 та x_2 мають нецілочислові значення. Дробова частина x_1 дорівнює $\frac{4}{5}$, а змінної x_2 – $\frac{2}{5}$. Тому для змінної x_1 $\left(\frac{4}{5} > \frac{2}{5}\right)$ складаємо додаткове обмеження. Із останньої симплекс-таблиці маємо

$$x_1 + \frac{3}{5}x_3 + \frac{2}{5}x_4 = \frac{9}{5}.$$

До системи обмежень задачі (2.13) – (2.15) додаємо нерівність

$$f(1)x_1 + f\left(\frac{3}{5}\right)x_3 + f\left(\frac{2}{5}\right)x_4 \geq f\left(\frac{9}{5}\right);$$

$$\frac{3}{5}x_3 + \frac{2}{5}x_4 \geq \frac{4}{5};$$

$$3x_3 + 2x_4 \geq 4.$$

Таким чином, обмеження задачі матимуть вигляд

$$\begin{cases} x_1 + \frac{3}{5}x_3 + \frac{2}{5}x_4 = \frac{9}{5} \\ x_2 - \frac{1}{5}x_3 + \frac{1}{5}x_4 = \frac{2}{5} \\ 3x_3 + 2x_4 \geq 4. \end{cases}$$

Відніmemo додаткову змінну $x_5 \geq 0$ в останній нерівності і помножимо одержану рівність на (-1) . Для розв'язання отриманої задачі потрібно застосувати двоїстий симплекс-метод.

i	Б	C_σ	P_0	1	-1	0	0	0
				P_1	P_2	P_3	P_4	P_5
1	P_1	1	$\frac{9}{5}$	1	0	$\frac{3}{5}$	$\frac{2}{5}$	0
2	P_4	-1	$\frac{2}{5}$	0	$\frac{2}{5}$	$-\frac{1}{5}$	$\frac{1}{5}$	0
3	P_5	0	-4	0	0	-3	-2	1
4			$\frac{7}{5}$	0	0	$\frac{4}{5}$	$\frac{1}{5}$	0

i	Б	C_σ	P_0	1	-1	0	0	0
				P_1	P_2	P_3	P_4	P_5
1	P_1	1	1	1	0	0	0	$\frac{1}{5}$
2	P_2	-1	0	0	1	$-\frac{1}{2}$	0	$\frac{1}{10}$
3	P_4	0	2	0	0	$\frac{3}{2}$	1	$-\frac{1}{2}$
4			1	0	0	$\frac{1}{2}$	0	$\frac{1}{10}$

Із таблиці видно, що вихідна задача цілочислового програмування має оптимальний план $X^* = (1; 0; 0; 2)$, при якому $F_{\max} = 1$.

2.3. Метод гілок і меж

Цей метод можна застосовувати для розв'язання як цілком, так і частково цілочислових задач дискретного програмування. Припустимо для визначеності, що модель має такий вигляд:

$$\sum_{j=1}^n c_j x_j \rightarrow \max \quad (2.17)$$

за наявності обмежень

$$\sum_{j=1}^n a_{i,j} x_j \leq b_i, \quad i=1, 2, 3, \dots, m$$

$$x_j - \text{цілі}, \quad j=1, 2, 3, \dots, p; \quad (p \leq n),$$

$$x_j \geq 0, \quad j=p+1, p+2, \dots, n.$$

Крім того, припустимо, що для кожної цілочислової змінної можливо задати верхню та нижню границі, в межах яких безумовно міститься її оптимальне значення:

$$L_j \leq x_j \leq U_j, \quad j=1, 2, 3, \dots, p. \quad (2.18)$$

Зазвичай $L_j=0$, але це не є обов'язковим, і умова $L_j=0$ не призводить до втрати загальності. Але, використовуючи символ L_j , простіше описати алгоритм.

Ідея методу гілок і меж ґрунтується на наступному елементарному факті. Розглянемо будь-яку змінну x_j і прийнемо, що I є деяке ціле число, де $L_j \leq I \leq U_j - 1$. Тоді оптимальний розв'язок задачі (2.17) буде задовольняти або лінійне обмеження $x_j \geq I+1$, або лінійне обмеження $x_j \leq I$.

Наприклад, маємо розв'язок задачі лінійного програмування (яке не враховує умову дискретності), в якому, наприклад, $x_1 = 5\frac{2}{3}$. Цієї задачі ставляться у відповідність дві нові задачі, в одній з яких міститься умова $6 \leq x_1 \leq U_1$, а в іншій $L_1 \leq x_1 \leq 5$. Далі припустимо, що кожна з цих задач має оптимальний розв'язок, який задовольняє умову дискретності. Тоді розв'язок, який доставляє більше значення цільової функції, є оптимальним. Зазвичай ці дві задачі не задовольняють умові дискретності, і тоді можуть знадобитися додаткові обчислення. Тобто метод гілок і меж є метаалгоритмом і оснований на розв'язанні множини задач лінійного програмування.

Алгоритм. На кожній ітерації (позначимо її m) є нижня границя (оцінка), скажемо x_0^m , оптимального значення цільової функції. Для спрощення викладання припустимо, що на першій ітерації x_0^1 або строго менше оптимальної цільової функції, або дорівнює значенню цільової функції, яке відповідає фіксованому припустимому розв'язку. За відсутності будь-якої інформації про особливості задачі можна прийняти $x_0^1 = -M$, де M - свідомо велике число. Окрім нижньої границі, маємо головний список задач лінійного програмування, які підлягають розв'язанню. Єдиною відмінністю цих задач одна від одної є зміна умов (2.2). На першій ітерації список містить лише одну задачу (2.17) – (2.18).

На довільній ітерації m виконуються наступні кроки:

Крок 1. Припинити обчислення, якщо головний список порожній. У протилежному випадку вибрати одну з задач лінійного програмування з головного списку.

Крок 2. Розв'язати обрану задачу. Якщо вона не має припустимого розв'язку або якщо отримане в результаті оптимальне значення цільової

функції менше або дорівнює x_0^m , то прийняти $x_0^{m+1} = x_0^m$ і повернутися до кроку 1. У протилежному випадку перейти до кроку 3.

Крок 3. Якщо отриманий оптимальний розв'язок задачі лінійного програмування задовольняє цілочислові обмеження, то зафіксувати (запам'ятати) його, прийняти x_0^{m+1} таким, що дорівнює відповідному оптимальному значенню цільової функції, і повернутися до кроку 1.

Крок 4. Вибрати будь-яку змінну x_j , $j=1, 2, 3, \dots, p$; ($p \leq n$), яка не приймає цілого значення в отриманому оптимальному розв'язку задачі лінійного програмування. Нехай b_j відповідає цьому значенню, а $[b_j]$ визначає найбільше ціле число, яке менше чи дорівнює b_j . Додати дві задачі лінійного програмування в головний список. Ці дві задачі ідентичні задачі, вибраної на кроці 1, за винятком того, що в одній з них нижню границю x_j замінено на $[b_j] + 1$, а в іншій верхню границю x_j замінено на $[b_j]$. Прийняти $x_0^{m+1} = x_0^m$ і повернутися до кроку 1.

Приклад. Розглянемо задачу

$$\begin{aligned} 3x_1 + 3x_2 + 13x_3 &\rightarrow \max, \\ -3x_1 + 6x_2 + 7x_3 &\leq 8, \\ 6x_1 - 3x_2 + 7x_3 &\leq 8, \end{aligned} \quad (2.19)$$

де кожна змінна x_j повинна бути невід'ємною та цілою. Припустимо, що задані границі на кожну змінну

$$0 \leq x_j \leq 5, \quad j=1, 2, 3. \quad (2.20)$$

Як було прийнято, позначимо x_0^m значення цільової функції. На першій ітерації приймемо нижню границю $x_0^1 = 0$, оскільки при всіх $x_j = 0$ маємо припустимий розв'язок. Головний список містить лише одну задачу лінійного програмування (2.19), яку назвемо задачею 1. Виберемо цю задачу на кроці 1 і знайдемо її оптимальний розв'язок (за допомогою симплекс-алгоритму) на кроці 2:

$$x_0 = 16, x_1 = x_2 = 2\frac{2}{3}, x_3 = 0 \text{ (задача 1)}.$$

Оскільки розв'язок не є цілочисловим, перейдемо від кроку 3 до кроку 4 і виберемо змінну x_1 . Внесемо до головного списку дві задачі:

- задачу 2, в якій обмеження (2.20) заміняються обмеженнями

$$3 \leq x_1 \leq 5, \quad 0 \leq x_2 \leq 5, \quad 0 \leq x_3 \leq 5;$$

- задачу 3, де обмеження (2.20) приймають вигляд

$$0 \leq x_1 \leq 2, \quad 0 \leq x_2 \leq 5, \quad 0 \leq x_3 \leq 5.$$

Повертаючись до кроку 1, при $x_0^2 = x_0^1 = 0$, вибираємо задачу 2. На

кроці 2 з'ясовується відсутність припустимого розв'язку задачі 2. Тому приймемо $x_0^3 = x_0^2 = 0$ і повернемося до кроку 1.

Виберемо задачу 3 і отримаємо оптимальний розв'язок задачі лінійного програмування:

$$x_0 = 15\frac{5}{7}, x_1 = x_2 = 2, x_3 = \frac{2}{7} \text{ (задача 3).}$$

Цей розв'язок не є цілочисловим. Тому перейдемо від кроку 3 до кроку 4, де виберемо змінну x_3 і внесемо до головного списку дві задачі:

- задачу 4, обмеження (2.20):

$$0 \leq x_1 \leq 2, 0 \leq x_2 \leq 5, 1 \leq x_3 \leq 5;$$

- задачу 5, з обмеженнями (2.20):

$$0 \leq x_1 \leq 2, 0 \leq x_2 \leq 5, 0 \leq x_3 \leq 0.$$

Відмітимо, що задачі 4 і 5 відрізняються від задачі 3 лише границями для x_3 .

Повертаючись до кроку 1, виберемо з головного списку задачу 4 і знайдемо її оптимальний розв'язок:

$$x_0 = 15, x_1 = x_2 = \frac{1}{3}, x_3 = 1 \text{ (задача 4).}$$

І переходимо до кроку 4. Виберемо змінну x_2 , внаслідок чого отримаємо ще дві задачі:

- задачу 6, з обмеженнями (2.20):

$$0 \leq x_1 \leq 2, 1 \leq x_2 \leq 5, 1 \leq x_3 \leq 5;$$

- задачу 7, з обмеженнями (2.20):

$$0 \leq x_1 \leq 2, 0 \leq x_2 \leq 0, 1 \leq x_3 \leq 5.$$

Таким чином, у головному списку є три задачі – задачі 5, 6 і 7. Повертаючись до кроку 1, виберемо задачу 6. На кроці 2 з'ясовується, що задача 6 не має припустимого розв'язку і тому треба повернутися до кроку 1, прийнявши $x_0^6 = 0$. Виберемо з головного списку задачу 7, оптимальний розв'язок якої має вигляд

$$x_0 = 14\frac{6}{7}, x_1 = x_2 = 0, x_3 = 1\frac{1}{7} \text{ (задача 7).}$$

Оскільки значення x_4 не є цілим, на кроці 4 отримаємо:

- задачу 8, з обмеженнями (2.20):

$$0 \leq x_1 \leq 2, 0 \leq x_2 \leq 0, 2 \leq x_3 \leq 5;$$

- задачу 9, з обмеженнями (2.20):

$$0 \leq x_1 \leq 2, 0 \leq x_2 \leq 0, 1 \leq x_3 \leq 1.$$

Задача 8 не має припустимого розв'язку. Задача 9 призводить до розв'язку

$$x_0 = 13, x_1 = x_2 = 0, x_3 = 1 \text{ (задача 9).}$$

Тому на кроці 2 фіксуємо цій розв'язок і приймаємо $x_0^9 = 13$. Повертаючись до кроку 1, знаходимо, що у головному списку залишилась лише одна задача 5. Оптимальний розв'язок цієї задачі лінійного програмування:

$$x_0 = 13, x_1 = 2, x_2 = 2\frac{1}{3}, x_3 = 0 \text{ (задача 5).}$$

Оскільки значення цільової функції x_0 на цій ітерації дорівнює $x_0^9 = 13$, тому, повертаючись до кроку 1, припиняємо обчислення.

Головний список пустий, а побудовані нові задачі на основі нецілочислового розв'язку задачі 5 будуть мати менші значення цільової функції, ніж отримане значення $x_0^9 = 13$ і тому не будуть оптимальними.

Слід відмітити, що можливість вільного вибору виникає в двох місцях: вибір задачі з головного списку на кроці 1 і вибір змінної, по якій формуються додаткові задачі на кроці 4. Число ітерацій, потрібних для розв'язання задачі, може суттєво змінюватися залежно від того, яким чином буде реалізовуватися вибір вказаних альтернатив.

Хід ітерацій можна подати графічно у вигляді дерева. Кожна вершина дерева відображає задачу з головного списку, а кожна гілка веде до однієї з задач, що додається до головного списку. Метод має також назву метода повернення. При обриві деяких гілок треба повернутися до початку, і проглянути відповідні гілки з метою пошуку будь-якої нерозв'язаної задачі, яка міститься в блок-схемі дерева.

2.4. Задача про призначення

Задачу про призначення можна коротко сформулювати так: задано n робіт, кожна з яких може виконати будь-який з n виконавців. Вартість виконання роботи i виконавцем j становить $t_{i,j}$. Потрібно розподілити виконавців по роботах, тобто призначити одного виконавця на кожну роботу таким чином, щоб мінімізувати загальні витрати.

Для математичної постановки задачі визначимо змінні, які описують задачу, таким чином:

$$x_{i,j} = \begin{cases} 1, & \text{якщо робота } i \text{ виконується виконавцем } j, \\ 0, & \text{в інших випадках,} \end{cases}$$

де $i = 1, 2, 3, \dots, n$ і $j = 1, 2, 3, \dots, n$.

Тоді оптимізаційна модель описується так:

$$L = \sum_{i=1}^n \sum_{j=1}^n t_{i,j} x_{i,j} \rightarrow \min, \quad i \neq j, \quad (2.21)$$

за наявності обмежень

$$\sum_{i=1}^n x_{i,j} = 1; \sum_{j=1}^n x_{i,j} = 1, \quad (2.22)$$

$$x_{i,j} = 0 \text{ або } x_{i,j} = 1.$$

Якщо кількість агентів і завдань однакові, а загальна вартість виконання всіх завдань дорівнює сумі витрат виконання окремих завдань, то задача називається лінійною задачею про призначення. Цей варіант задачі є базовим і найпростішим. Зазвичай, говорячи про задачу про призначення без додаткових вимог, мають на увазі лінійну задачу про призначення. В інших варіантах задачі можуть бути додаткові умови, інші способи розрахунку загальних витрат, базові умови можуть бути дещо змінені. Наприклад, може бути неоднакова кількість агентів і завдань, нелінійність у визначенні загальних витрат тощо. В таких випадках кажуть про узагальнену задачу про призначення.

2.5. Задача комівояжера

В заданій мережі, кожній дузі відповідає деяка відстань або час руху між її вузлами (місцями). Треба знайти найкоротший час, за яким комівояжер планує відвідати всі n вузлів мережі, відвідати кожне місце по одному разу і повернутися в точку старту. Математична постановка задачі комівояжера зводиться до знаходження $n(n-1)$ булевих змінних $x_{i,j} \in \{0;1\}$, $(i, j = 1, 2, \dots, n; i \neq j)$, які мінімізують час відвідування комівояжером усіх заданих точок, тобто забезпечують

$$L = \sum_{i=1}^n \sum_{j=1}^n t_{i,j} x_{i,j} \rightarrow \min, \quad i \neq j,$$

де $t_{i,j}$ – це час руху (або відстань) з точки з номером i до точки з номером j . Якщо $x_{i,j} = 1$, то рухаючись по маршруту, комівояжер з точки i досягає безпосередньо точку j . Якщо $x_{i,j} = 0$, то при русі по маршруту рух з точки з номером i в точку з номером j не передбачений. При цьому мають бути виконані обмеження

$$\sum_{i=1}^n x_{i,j} = 1; \sum_{j=1}^n x_{i,j} = 1.$$

Дані обмеження відображають умови, за якими комівояжер повинен побувати в кожній з заданих точок лише один раз.

Крім того, додається ще одна умова - **розв'язок має бути циклом**. Ця умова суттєво відрізняє і ускладнює задачу комівояжера від задачі про призначення.

Як відомо, постановка задачі є задачею про призначення, розв'язок якої містить n змінних, що дорівнюють одиниці, а решта $n(n-1)$ є нульовими. Розв'язок може складатися з декількох простих вершинно-непересічних циклів, так званих підциклів, що проходять через менше ніж n число точок. Тому, для отримання маршруту, що зв'язує n точок, необхідно доповнити наведені вище обмеження умовами, що забезпечують зв'язність шуканого циклу. Існує декілька способів застосування методу гілок і меж для розв'язання задачі комівояжера. Одні підходи засновані на додаванні нових умов до задачі, які забороняють підцикли. Іншим можливим підходом є зміна значень часу пересування між вузлами (або відстані між вузлами). Тобто на кожній ітерації відповідним чином треба змінювати цільову функцію.

Розглянемо алгоритм, який має назву «метод виключення підциклів». На черговій ітерації k виконуються такі кроки:

Крок 1. Припинити розрахунки, якщо основний список задач пустий. У протилежному випадку взяти одну з задач (матрицю $\{t_{i,j}\}$), виключивши її з основного списку.

Крок 2. Розв'язати відповідну задачу бінарного програмування (2.21) за умов (2.22). Якщо отримане оптимальне значення цільової функції більше чи дорівнює поточному значенню цільової функції L , то повернутися до кроку 1, не змінюючи L . (Початкове значення L на першій ітерації може дорівнювати будь-якому свідомо великому числу). У протилежному випадку перейти до кроку 3.

Крок 3. Якщо отриманий оптимальний розв'язок задачі (2.21) - (2.22) є циклом, то зафіксувати його, присвоїти L значення цільової функції, яке отримане на даній ітерації, і повернутися до кроку 1. Якщо отриманий розв'язок не є циклом – перейти до кроку 4.

Крок 4. В отриманому розв'язку знайти найменший підцикл, тобто підцикл, який містить найменшу кількість відрізків. Кожному $x_{i,j}=1$, яке входить до цього підциклу, поставити у відповідність задачу бінарного програмування, додавши її до основного списку і прийняв відповідне значення в матриці часу перельотів $t_{i,j} = \infty$ (або свідомо великому числу). А всі інші коефіцієнти залишити незмінними, тобто такими ж, як і в задачі, вибраній на кроці 1. Перейти до кроку 1.

Такий алгоритм містить вільний вибір задачі зі списку, що призводить до різного ходу розв'язання задачі. Тобто кількість ітерацій при розв'язанні однієї задачі може істотно відрізнятись. Але було доведено, що алгоритм завжди приводить до побудови оптимального розв'язку.

У деяких випадках виникає задача побудувати найкоротший маршрут відвідання системи точок, в якій точка старту і точка завершення маршруту не співпадають, тобто маршрут є розімкненим. Розв'язання цієї задачі

можливо за допомогою зведення її до класичної задачі комівояжера. Нехай, наприклад, точкою старту є вузол з номером s , а точкою фінішу є точка з номером f . У цьому разі рух починається з точки s і завершується точкою f . Щоб класичний алгоритм виконав розрахунок маршруту, необхідно у матриці t_{ij} відповідний час $t_{f,s}$ дорівняти нулю. Крім того, необхідно заборонити всі маршрути, які виходять з точки f , для чого час відповідних маршрутів призначається достатньо великим. І також необхідно заборонити всі маршрути, які входять до вузла s . Тобто надаємо таких значень елементам матриці:

$$t_{f,s} = 0; \quad t_{f,i} = \infty, \quad i = 1, \dots, N, \quad i \neq s; \quad t_{i,s} = \infty, \quad i = 1, \dots, N, \quad i \neq f;$$

Ці параметри забезпечують обов'язковий старт з точки s , рух до точки f та замикання маршруту через відрізок (f,s) , який дорівнює нулю. Це ілюструє рис. 2.2.

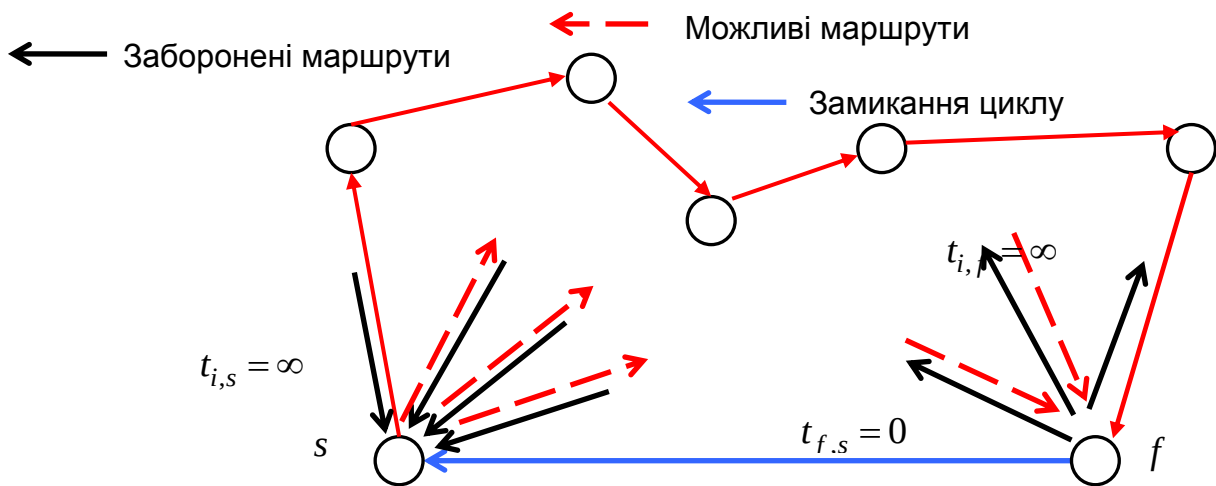


Рис. 2.2. Схема замикання маршруту з фіксованою точкою старту і посадки

Приклад 2.3. Розглянемо задачу комівояжера, яка задана матрицею, що наведена в табл. 2.2.

Таблиця 2.2

Вузли	1	2	3	4	5
1	∞	10	25	25	10
2	1	∞	10	15	2
3	8	9	∞	20	10
4	14	10	24	∞	15
5	10	8	25	27	∞

Діагональні елементи дорівнюють нескінченності для того, щоби при розв'язанні задачі заборонити маршрути із пункту i у пункт i . На практиці

замість нескінченності можна задати відповідні елементи, що дорівнюють достатньо великому числу, у цьому випадку, наприклад, 1000. Це знеможливіть потрапляння відповідних змінних до оптимального розв'язку.

На першій ітерації приймемо $x_0^1 = 65$, що відповідає циклічному маршруту $x_{1,2} = x_{2,3} = x_{3,4} = x_{4,5} = x_{5,1} = 1$. Головний список містить лише одну задачу о призначеннях, в якій значення $t_{i,j}$ відповідають значенням, наведеним у табл. 2.1. Тому вибираємо цю задачу. Знаходимо, що оптимальний розв'язок задачі дискретного програмування (2.21) - (2.22) має вигляд $x_{1,5} = x_{5,1} = x_{2,3} = x_{3,4} = x_{4,2} = 1$, тобто містить два підцикли і дає значення цільової функції 60. Переходимо до кроку 4. Вибираємо підцикл $x_{1,5} = x_{5,1} = 1$, тому що до нього входить менша кількість вузлів. Вносимо до головного списку дві задачі, покладаючи в однієї з них $t_{1,5} = \infty$, а в іншій $t_{5,1} = \infty$, залишивши всі інші значення $t_{i,j}$ без змін. Приймаємо $x_0^2 = x_0^1 = 65$.

Обґрунтувати крок 4 нескладно. Оскільки $x_{1,5} = x_{5,1} = 1$ не припустимий, то в оптимальному розв'язку буде або $x_{1,5} = 0$, або $x_{5,1} = 0$ (чи обидві змінні будуть дорівнювати нулю). Задачі, поставлені на кроці 4, припускають обидві з цих можливостей. Аналогічний хід міркувань справедливий для будь-якого підциклу. Однак у ході роботи цього алгоритму можливо включення до головного списку тотожних задач. Тому для задач великих розмірностей доцільно включити до алгоритму перевірку задач, які включаються до головного списку на тотожність з внесеними до списку, хоча для задач середніх розмірностей може статися доцільним розв'язання надлишкових задач, а не перевірка повторень.

Повертаючись до кроку 1, виберемо задачу, в якій $t_{1,5} = \infty$, назвавши її задачею 2. Оптимальне значення цільової функції, яке відповідає цій задачі о призначеннях, дорівнює 65. Це значення являє собою нижню границю для будь-якого підциклу, який виключає маршрут із вузла 1 до вузла 5. Оскільки цикл довжиною 65 вже існує, немає необхідності далі розглядати цю задачу. Таким чином, за кроком 2 приймаємо $x_0^3 = x_0^2 = 65$ і повертаємось до кроку 1. Виберемо ту задачу, що залишилась, в якій $t_{5,1} = \infty$, і назвемо її задачею 3. Далі отримаємо оптимальний розв'язок задачі, тобто значення цільової функції дорівнює 62, і переходимо до кроку 3. Розв'язок $x_{1,5} = x_{5,2} = x_{2,3} = x_{3,4} = x_{4,1} = 1$ є циклом. Тому фіксуємо розв'язок і покладаємо $x_0^4 = 62$ і повертаємось до кроку 1. Головний список пустий і розрахунки припиняємо.

Підсумовуючи, відмітимо, що симплекс-метод лінійного програмування служить основою для розв'язання задач дискретного програмування, яке, в свою чергу, входить як процедура до задачі комівояжера.

3. ТРАНСПОРТНА ЗАДАЧА

3.1. Постановка транспортної задачі

Симплексний метод є універсальним способом розв'язування задач лінійного програмування з неперервними аргументами. Проте окремі типи задач лінійного програмування мають таку структуру, яка дає змогу побудувати методи розв'язування значно простіші за симплексний. Найважливішим типом таких задач є транспортна задача (ТЗ).

До транспортної задачі можуть зводитися інші задачі дослідження операцій, які на перший погляд не мають нічого спільного з перевезеннями. Такими задачами є задачі оптимального розміщення виробництва, баз і складів; оптимальний добір у племянній справі тощо.

Постановка транспортної задачі. Нехай у m пунктах $A_1, A_2, \dots, A_m$ зосереджено однорідний вантаж у кількостях a_i ($i = \overline{1, m}$), який необхідно цілковито розвезти у n інших пунктів $B_1, B_2, \dots, B_n$ у кількостях b_j ($j = \overline{1, n}$). Очевидно, що $a_i > 0$, $i = \overline{1, m}$; $b_j > 0$, $j = \overline{1, n}$. Відомо, що вартість перевезення одиниці вантажу з пункту A_i у пункт B_j дорівнює c_{ij} . Необхідно мінімізувати загальну вартість перевезень за умови цілковитого вивезення вантажу з пунктів відправки і розвезення його у потрібних кількостях до пунктів призначення.

Побудуємо математичну модель ТЗ. Нехай x_{ij} – кількість одиниць вантажу, які необхідно перевезти з пункту A_i до пункту B_j . Тоді:

$$z = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \min, \quad (3.1)$$

$$\sum_{j=1}^n x_{ij} = a_i; \quad i = \overline{1, m}, \quad (3.2)$$

$$\sum_{i=1}^m x_{ij} = b_j, \quad j = \overline{1, n}. \quad (3.3)$$

Умова (3.2) гарантує цілкове вивезення вантажу з усіх пунктів, а (3.3) – означає цілкове задоволення попиту в усіх пунктах призначення. Умови (3.2) і (3.3) називають основними обмеженнями ТЗ. Очевидна також необхідність накладення на змінні задачі природних обмежень невід'ємності:

$$x_{ij} \geq 0, \quad i = \overline{1, m}; \quad j = \overline{1, n}, \quad (3.4)$$

оскільки перевезення з пунктів B_j у пункти A_i не допускаються. Задачу (3.1) – (3.4) називають транспортною задачею за критерієм вартості перевезень.

Отже, ТЗ є задачею лінійного програмування з mn числом змінних; $(m+n)$ обмеженнями-рівностями та mn обмеженнями-нерівностями. Матрицю $C = (c_{ij})_{i=1, \dots, m; j=1, \dots, n}$ називають матрицею вартості перевезень. Вектор $\bar{a} = (a_1, a_2, \dots, a_m)^T$ називають вектором запасів, а вектор $\bar{b} = (b_1, b_2, \dots, b_n)^T$ – вектором потреб.

Для існування розв'язку ТЗ важливим питанням є співвідношення між сумою запасів $\sum_{i=1}^m a_i$ і сумою потреб $\sum_{j=1}^n b_j$. Якщо сума запасів дорівнює сумі потреб, тобто

$$\sum_{i=1}^m a_i = \sum_{j=1}^n b_j, \quad (3.5)$$

то транспортну задачу називають закритою. Співвідношення (3.5) називають умовою балансу.

У разі порушення співвідношення (3.5) отримуємо транспортну задачу з порушеним балансом (або відкриту ТЗ).

3.2. Властивості закритої транспортної задачі

Закрита ТЗ має деякі важливі властивості, які дають змогу спростити її розв'язання порівняно із симплекс-методом:

1. Кожну змінну x_{ij} ТЗ налічує система основних обмежень лише двічі: вперше у (3.2), а вдруге – у (3.3).

2. Елементи матриці основних обмежень – це нулі та одиниці. Кожен стовпець матриці має лише дві одиниці, решта елементів – нулі. Кожен рядок матриці має n або m одиниць, решта елементів – нулі. Запишемо, наприклад, (3.1) – (3.3) для $m=2$ і $n=3$:

$$z = c_{11}x_{11} + c_{12}x_{12} + c_{13}x_{13} + c_{21}x_{21} + c_{22}x_{22} + c_{23}x_{23} \rightarrow \min;$$

$$\begin{cases} x_{11} + x_{12} + x_{13} = a_1, \\ x_{21} + x_{22} + x_{23} = a_2; \end{cases} \quad \begin{cases} x_{11} + x_{21} = b_1, \\ x_{12} + x_{22} = b_2, \\ x_{13} + x_{23} = b_3; \end{cases}$$

Матрична форма обмежень:

$$\begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_{11} \\ x_{12} \\ x_{13} \\ x_{21} \\ x_{22} \\ x_{23} \end{pmatrix} = \begin{pmatrix} a_1 \\ a_2 \\ b_1 \\ b_2 \\ b_3 \end{pmatrix}.$$

Систему лінійних алгебричних рівнянь (3.2) і (3.3), яка задає основні обмеження ТЗ, запишемо у матричній формі $A\bar{X} = \bar{B}$, де $\bar{X} = (x_{11}, x_{12}, \dots, x_{1n}; x_{21}, x_{22}, \dots, x_{2n}; \dots; x_{m1}, x_{m2}, \dots, x_{mn})^T$ – вектор-стовпець невідомих, що має mn компонент; $\bar{B} = (a_1, a_2, \dots, a_m; b_1, b_2, \dots, b_n)^T$ – вектор-стовпець обмежень;

$\bar{A} = (\bar{P}_{11}, \bar{P}_{12}, \dots, \bar{P}_{1n}; \bar{P}_{21}, \bar{P}_{22}, \dots, \bar{P}_{2n}; \dots; \bar{P}_{m1}, \bar{P}_{m2}, \dots, \bar{P}_{mn})$ – матриця, в якій

$\bar{P}_{ij} = (\underbrace{0, 0, \dots, 1, 0, \dots, 0}_{m+j}, \underbrace{1, 0, \dots, 0}_{n-j})^T$ – вектор перевезень вантажу з пункту

$A_i (i = \overline{1, m})$ у пункт $B_j (j = \overline{1, n})$, який відповідає невідомому x_{ij} .

3. Теорема 3.1. Ранг матриці A системи лінійних алгебричних рівнянь (3.2) і (3.3), яка задає основні обмеження закритої ТЗ, дорівнює $m + n - 1$.

4. Теорема 3.2. Для розв'язання ТЗ необхідно і достатньо, щоб виконувалася умова балансу (3.5).

З теореми 3.2 випливає, що побудувати розв'язок можна тільки для закритої ТЗ. Для розв'язання відкритої ТЗ її попередньо необхідно звести до закритої. Проілюструємо це.

Нехай $\sum_{i=1}^m a_i < \sum_{j=1}^n b_j$ (перевищення потреб над запасами). У цьому випадку вводять фіктивного постачальника A_{m+1} з ресурсом $a_{m+1} = \sum_{j=1}^n b_j - \sum_{i=1}^m a_i$.

Нехай $\sum_{i=1}^m a_i > \sum_{j=1}^n b_j$ (перевищення запасів над потребами). У цьому випадку вводять фіктивного споживача B_{n+1} з ресурсом $b_{n+1} = \sum_{i=1}^m a_i - \sum_{j=1}^n b_j$.

Вартість перевезення від фіктивного постачальника A_{m+1} (або до фіктивного споживача B_{n+1}) вважають такою, що дорівнює нулю.

3.3. Властивості опорних планів транспортної задачі

З теореми 3.1 попереднього параграфа випливає, що серед $m + n$ лінійно залежних рівнянь системи (3.2), (3.3) з mn невідомими можна виокремити підсистеми, які містять $m + n - 1$ лінійно незалежних рівнянь.

Виокремлюючи у таких підсистемах базисні невідомі, можна знайти базисні допустимі розв'язки (або опорні плани) $\bar{X}_{on}$ системи (3.2) –

(3.4). Кожен опорний план має не більше $m+n-1$ ненульових (додатних) компонент і не менше $mn-(m+n-1)=(m-1)(n-1)$ нульових.

Опорний план ТЗ (вектор $\overline{X}_{on}$), який містить $m+n-1$ ненульових компонент, називають не виродженим. У виродженого опорного плану кількість ненульових компонент є меншою за $m+n-1$.

Кожній компоненті $x_{i_0 j_0}$ опорного плану $\overline{X}_{on}$ взаємно-однозначно відповідає вектор перевезень $\overline{P}_{i_0 j_0}$ вантажу з пункту A_{i_0} до пункту B_{j_0} . Нехай E_{on} – система векторів перевезень, що відповідають компонентам $\overline{X}_{on}$. Очевидно, що E_{on} є системою лінійно незалежних векторів.

Під час обчислення планів ТЗ зручно користуватися прямокутною таблицею розміром $m \times n$. Клітинку таблиці, яка розміщена на перетині i -го рядка та j -го стовпця, позначимо (i, j) . У клітинці (i, j) записують значення c_{ij} , x_{ij} і інших величин. Нехай символ « $\leftrightarrow$ » позначає взаємно-однозначну відповідність, тоді

$$(i, j) \leftrightarrow x_{ij}; \quad x_{ij} \leftrightarrow \overline{P}_{ij} \Rightarrow (i, j) \leftrightarrow \overline{P}_{ij}.$$

Взаємно-однозначна відповідність $x_{ij} \leftrightarrow \overline{P}_{ij}$ дає змогу інтерпретувати дії над векторами E_{on} як певні дії над клітинками таблиці. Ця інтерпретація полегшує розуміння алгоритмів знаходження опорного та оптимального планів ТЗ.

Клітинку (i, j) зв'язують також з комунікацією $A_i B_j$ (шляхом) між пунктом постачання A_i та пунктом споживання B_j . Якщо переміщатися від A_i до B_j , то такому переміщенню відповідає вектор $\overline{P}_{ij}$, переміщенню від B_j до A_i відповідає вектор $-\overline{P}_{ij}$. Набір клітин $(i_1, j_1), (i_2, j_1), (i_2, j_2), (i_3, j_2), \dots$ або $(i_1, j_1), (i_1, j_2), (i_2, j_2), (i_2, j_3), \dots$ називають ланцюжками переміщень (просто ланцюжками). Кожна пара клітинок ланцюжка міститься або в одному рядку, або в одному стовпці таблиці так, що деякі три чи більше клітинок набору не є клітинками одного рядка або одного стовпця.

Перший ланцюжок задає такі переміщення між пунктами (символ « $\leftrightarrow$ » позначає напрям переміщення):

$$A_{i_1} \rightarrow B_{j_1} \rightarrow A_{i_2} \rightarrow B_{j_2} \rightarrow A_{i_3} \rightarrow \dots$$

Другий ланцюжок задає такі переміщення між пунктами:

$$B_{j_1} \rightarrow A_{i_1} \rightarrow B_{j_2} \rightarrow A_{i_2} \rightarrow B_{j_3} \rightarrow \dots$$

Кожному ланцюжку переміщень відповідає певна лінійна комбінація векторів перевезень: $\overline{P}_{i_1 j_1} - \overline{P}_{i_2 j_1} + \overline{P}_{i_2 j_2} - \overline{P}_{i_3 j_2} + \dots$ (для першого ланцюжка) і $-\overline{P}_{i_1 j_1} + \overline{P}_{i_1 j_2} - \overline{P}_{i_2 j_2} + \overline{P}_{i_2 j_3} - \dots$ (для другого ланцюжка).

Ланцюжок вигляду $(i_1, j_1), (i_2, j_1), (i_2, j_2), (i_3, j_2), \dots, (i_s, j_s), (i_1, j_s)$, або $(i_1, j_1), (i_1, j_2), (i_2, j_2), (i_2, j_3), \dots, (i_k, j_k), (i_k, j_1)$, є замкнутим. Замкнутий ланцюжок називають циклом.

Перший замкнутий ланцюжок задає такі переміщення:

$$A_{i_1} \rightarrow B_{j_1} \rightarrow A_{i_2} \rightarrow B_{j_2} \rightarrow A_{i_3} \rightarrow \dots \rightarrow A_{i_s} \rightarrow B_{j_s} \rightarrow A_{i_1}.$$

Другий замкнутий ланцюжок задає такі переміщення:

$$B_{j_1} \rightarrow A_{i_1} \rightarrow B_{j_2} \rightarrow A_{i_2} \rightarrow B_{j_3} \rightarrow \dots \rightarrow B_{j_k} \rightarrow A_{i_k} \rightarrow B_{j_1}.$$

В обох випадках кінцевий пункт переміщення збігається з початковим пунктом. А це означає, що відповідна лінійна комбінація векторів переміщень дорівнює $\bar{0}$: $\bar{P}_{i_1 j_1} - \bar{P}_{i_2 j_1} + \bar{P}_{i_2 j_2} - \bar{P}_{i_3 j_2} + \dots + \bar{P}_{i_s j_s} - \bar{P}_{i_1 j_s} = \bar{0}$ (для першого ланцюжка) і $-\bar{P}_{i_1 j_1} + \bar{P}_{i_1 j_2} - \bar{P}_{i_2 j_2} + \bar{P}_{i_2 j_3} - \dots - \bar{P}_{i_k j_k} + \bar{P}_{i_k j_1} = \bar{0}$ (для другого ланцюжка). Отже, система векторів перевезень, яка відповідає замкнутому ланцюжку, є лінійно залежною.

Приклад 3.1. Позначити ланцюжки $\{(2,1), (5,1), (5,4), (4,4), (3,4), (3,2)\}$ (позначення символом «•») і $\{(3,1), (1,1), (1,3), (2,3), (2,4)\}$ (позначення символом «*») у прямокутній таблиці розміром 5×4 .

i \ j	1	2	3	4
1				
2	•		•	
3				
4			•	•
5	•			•

Нехай E – деяка система векторів перевезень $\bar{P}_{ij}$, а I – множина пар індексів (i, j) , які відповідають векторам системи E .

Теорема 3.3. Система векторів перевезень E лінійно незалежна тоді і тільки тоді, коли з клітин, які відповідають векторам системи E , не можна скласти замкнутий ланцюжок (цикл).

Якщо опорний план $\bar{X}_{on}$ записати у вигляді таблиці перевезень, то вона має не більше як $m+n-1$ клітинок, у яких знаходяться додатні значення. Ці клітинки відповідають базисним змінним, отже їх називають базисними клітинками, а решту – вільними. Справедливий такий наслідок з теореми 3.3.

Наслідок 3.1 (з теореми 3.3). Для того, щоб деякий план у таблиці перевезень був опорним, необхідно і достатньо, щоб із базисних клітинок не можна було скласти замкнутий ланцюжок (іншими словами: з комунікацій, які відповідають базисним клітинкам, не можна було б скласти замкнутий маршрут).

Приклад 3.2

i \ j	1	2	3	4
1	2	0	0	5
2	0	3	3	0
3	1	0	0	3

1. План не є опорним, адже існує цикл $(1,1),(1,4),(3,4),(3,1)$.

За великих розмірів таблиці перевезень візуальний пошук замкнутих ланцюжків (циклів) є досить складним. Можна алгоритмізувати пошук циклів за допомогою методу викреслювання.

Нехай R – множина ненульових елементів у таблиці перевезень T . Визначити наявність циклів у таблиці T .

Алгоритм методу викреслювання:

1. Переглядаємо усі рядки таблиці T і викреслюємо ті з них, які не мають елементів R або мають тільки один елемент R .
2. Переглядаємо усі стовпці таблиці T і викреслюємо ті з них, які не мають елементів R або мають тільки один елемент R . Раніше викреслені елементи до уваги не беруть.

Повторюємо кроки 1 – 2. Після скінченної кількості кроків процес викреслювання завершиться. Якщо усі рядки (стовпці) таблиці T викреслені, то побудувати цикл не можна (план опорний). Якщо залишилася підтаблиця, у кожному рядку і стовпці якої є не менше двох елементів R , то R містить цикл (цикли), і відповідний план не є опорним.

Приклад 3.3. За допомогою методу викреслювання встановити, чи вказаний план є опорним (див. приклад 3.2). Нумери ліній вказують на порядок викреслення.

i \ j	1	2	3	4
1	2	0	0	5
2	0	3	3	0
3	1	0	0	3

1 2

1. План не є опорним, адже існує цикл $(1,1),(1,4),(3,4),(3,1)$.

i \ j	1	2	3	4
1	2	3	0	0
2	1	0	1	0
3	0	0	0	3

5 1 2 3

1. Опорний план.

Нехай E – деяка лінійно незалежна система векторів перевезень $\bar{P}_{ij}$, а $\bar{P}_{kl}$ – вектор, що не входить до E (тобто $\bar{P}_{kl} \notin E$).

Теорема 3.4. Для того, щоб вектор $\bar{P}_{kl} \notin E$ був лінійною комбінацією векторів системи E , необхідно і достатньо, щоб з клітинок, які

відповідають векторам системи E , можна було скласти замкнутий ланцюжок (цикл), що замикається на клітинці (k,l) .

Як відомо з лінійної алгебри, довільний вектор деякого векторного простору однозначно розкладається через вектори базису цього простору. Отож вводимо поняття базису опорного плану.

Базисом опорного плану $\bar{X}_{on}$ ТЗ вважатимемо будь-яку систему з $m+n-1$ лінійно незалежних векторів перевезень $\bar{P}_{ij}$, яка містить усі вектори, що відповідають додатнім перевезенням.

Якщо $\bar{X}_{on}$ – не вироджений, то його базис визначають однозначно. Якщо $\bar{X}_{on}$ – вироджений, то його базис визначають неоднозначно (містить усі вектори, які відповідають додатнім перевезенням, й окремі вектори, які відповідають нульовим перевезенням).

Нехай B – множина клітинок, які відповідають векторам перевезень $\bar{P}_{ij}$ базису деякого опорного плану $\bar{X}_{on}$ ТЗ, а клітинка $(k,l) \in B$. Справедливий такий наслідок з теореми 3.4.

Наслідок 3.2 (з теореми 3.3). З клітинок множини B можна скласти єдиний ланцюжок, який замикається на клітинці (k,l) .

Єдиність існування ланцюжка, який замикається на клітинці (k,l) , випливає з того, що вектор $\bar{P}_{kl}$ єдиним чином розкладається через вектори базису опорного плану.

3.4. Алгоритми побудови опорних планів транспортної задачі

Як і в звичайному симплексному методі, розв'язання ТЗ полягає у переборі і перевірці на оптимальність опорних планів задачі, для чого треба знайти початковий опорний план.

Побудова опорного плану ТЗ методом північно-західного кута

Назва методу пов'язана з тим, що на кожному кроці визначається величина перевезення для верхньої лівої клітинки (північно-західної) незаповненої прямокутної таблиці перевезень.

На початку першого кроку таблиця перевезень є незаповненою, окрім стовпця запасів та рядка потреб. Для першої верхньої лівої клітинки (1,1) розглядаємо величини a_1 і b_1 . Можливі випадки:

- $a_1 > b_1$ – потребу b_1 можна задовольнити за рахунок запасу a_1 ($x_{11} = b_1$), отож з інших пунктів A_i ($i = 2, 3, \dots, m$) до пункту B_1 завозити нічого не потрібно ($x_{21} = x_{31} = \dots = x_{m1} = 0$) – викреслюємо перший стовпець. На місці a_1 запишемо $a_1' = a_1 - b_1$.

- $a_1 < b_1$ – увесь запас a_1 вивозимо в пункт B_1 ($x_{11} = a_1$), задовольняючи лише частину його потреб. Тоді $x_{12} = x_{13} = \dots = x_{1n} = 0$ – викреслюємо перший рядок. на місці b_1 запишемо $b_1' = b_1 - a_1$.
- $b_1 = a_1 = x_{11}$ – потреба і запас збігаються. Тоді $x_{21} = x_{31} = \dots = x_{m1} = x_{12} = x_{13} = \dots = x_{1n} = 0$ – викреслюємо перший стовпець і перший рядок.

Отже, в усіх випадках приймаємо $x_{11} = \min(a_1; b_1)$.

Далі розглянемо незаповнену частину таблиці, яка також є прямокутною. Для цієї незаповненої таблиці аналізуємо значення запасів/потреб верхньої лівої клітинки (на другому кроці це може бути клітинка (1,2), або клітинка (2,1), або клітинка (2,2)) за аналогією з першим кроком. Через обмеженість числа елементів a_i ($i = \overline{1, m}$) і b_j ($j = \overline{1, n}$) описаний алгоритм є скінченним. Отриманий план є опорним, оскільки з його клітинок, заповнених додатними перевезеннями, згідно з побудовою, не можна скласти замкнутий ланцюжок.

Домовимося нульові перевезення у таблиці не вказувати (залишати клітинки незаповненими), викреслювання здійснювати уявно, а старі та нові значення запасів/потреб відокремлювати у відповідному стовпці/рядку через косі риски.

Приклад 3.4. Знайти початковий опорний план ТЗ, заданий таблицею, в якій вказано вартості перевезень, запаси (a_i) і потреби (b_j).

i \ j	1	2	3	4	a_i
1	8	5	13	12	15
2	6	11	15	18	25
3	2	9	11	16	10
b_j	5	15	15	15	$\sum = 50$

Вартості перевезень при побудові опорного плану ТЗ методом північно-західного кута не мають значення, отож їх при побудові опорного плану не вказуватимемо.

i \ j	1	2	3	4	a_i
1	5	10			15/10
2		5	15	5	25/20/5
3				10	10
b_j	5	15/5	15	15/10	$\sum = 50$

Сірим кольором позначено базисні клітинки (6 клітинок). Отриманий опорний план є не виродженим ($6 = 3 + 4 - 1$). Сумарна вартість перевезення $z = 5 \cdot 8 + 10 \cdot 5 + 5 \cdot 11 + 15 \cdot 15 + 5 \cdot 18 + 10 \cdot 16 = 620$.

Опорний план ТЗ обов'язково має в кожному рядку і в кожному стовпці таблиці перевезень хоча б один додатний елемент, адже $a_i \neq 0 (i = \overline{1, m})$ і $b_j \neq 0 (j = \overline{1, n})$, отож найменше можливе число додатних змінних x_{ij} дорівнює $N = \max(m, n)$. На кожному кроці методу знаходимо лише один елемент $x_{ij} > 0$, а на місці хоча б одного з елементів a_i чи b_j виникає нуль. На останньому кроці на місці a_m і b_n нуль виникає одночасно. Отже, найбільше можливе число додатних змінних x_{ij} дорівнює $N = m + n - 1$.

Якщо $N < m + n - 1$, то опорний план є виродженим. Причиною виродженості опорного плану є рівність потреби і запасу на проміжному (не останньому) кроці алгоритму, адже у цьому випадку одночасно викреслюються і рядок, і стовпець. Число базисних змінних, які дорівнюватимуть нулю, збігатиметься з числом цих випадків.

Для вироджених опорних планів базисні змінні формують так: додатні змінні x_{ij} доповнюють $(m + n - 1) - N$ нульовими змінними x_{ij} так, щоб вони разом з додатними змінними не утворювали циклів. Домовимося надалі вказувати у таблиці нульові перевезення тільки для базисних змінних.

Приклад 3.5. Знайти початковий опорний план ТЗ, для якої вектор $\vec{a} = (1, 2, 3, 4, 5)^T$, а вектор $\vec{b} = (2, 1, 6, 6)^T$.

i \ j	1	2	3	4	a_i
1	1				1
2	1	1			2/1
3		0	3		3
4			3	1	4/1
5				5	5
b_j	2/1	1	6/3	6	$\sum = 15$

Додатні перевезення плану містяться у 7-ми клітинках. Отриманий опорний план є виродженим ($7 < 5 + 4 - 1$). Отож до числа базисних клітинок додамо клітинку (3, 2) з нульовим перевезенням.

При побудові ОП методом північно-західного кута не беруть до уваги вартості перевезень, отож опорний план може надто відрізнятись від оптимального плану.

Побудова опорного плану ТЗ методом мінімального елемента

Метод мінімального елемента дає змогу побудувати відразу достатньо економічний опорний план, що скорочує загальне число ітерацій при його наступній оптимізації (порівняно з методом північно-західного

кута). Це досягається тим, що в ОП вибирають такі додатні перевезення, яким відповідає найменша вартість.

Користуючись матрицею перевезень, на першому кроці визначаємо таке значення x_{ij} , для якого величина c_{ij} є мінімальною, тоді

$x_{ij} = \min(a_i, b_j)$. Можливі три випадки:

- якщо $a_i < b_j$, то решту клітинок i -го рядка заповнюємо нулями;
- якщо $a_i > b_j$, то решту клітинок j -го стовпця заповнюємо нулями;
- якщо $a_i = b_j$, то решту клітинок i -го рядка та j -го стовпця заповнюємо нулями.

Далі цей процес повторюємо з незаповненою частиною таблиці. Отриманий план є опорним, оскільки з його клітинок, заповнених додатними перевезеннями, не можна скласти цикл.

Домовимося вартість перевезень (c_{ij}) у таблиці вказувати у правому верхньому куті клітинки, а величини перевезень (x_{ij}) – у нижньому лівому куті; замість номерів рядків/стовпців безпосередньо вказуватимемо величини запасів/потреб відповідного пункту.

Приклад 3.6. Знайти опорний план ТЗ з прикладу 3.4 методом мінімального елемента.

$a_i \backslash b_j$	5	15	15/10	15
15	8	5	13	12
25/15	6	11	15	18
10/5	2	9	11	16

Послідовність заповнення клітинок: $(3,1), (1,2), (3,3), (3,2), (2,4)$. Отриманий опорний план є виродженням ($5 < 3 + 4 - 1$); до числа базисних додамо клітинку $(1,4)$. Сумарна вартість перевезення: $z = 5 \cdot 2 + 15 \cdot 5 + 10 \cdot 15 + 5 \cdot 11 + 15 \cdot 18 = 560$ ($z = 620$ – у прикладі 3.4).

Побудова опорного плану ТЗ методом Фоґеля

На довільному кроці методу:

- для кожного рядка та стовпця обчислюємо різницю (штраф) між значеннями найменшої вартості та вартості, наступної за величиною;
- обчислені штрафи записуємо у додаткові стовпці та рядки;
- виокремлюємо рядок чи стовпець з найбільшим штрафом (якщо їх декілька, то вибираємо довільний з них);

- у відокремленому рядку/стовпці вибираємо клітинку з найменшою вартістю;
- для вибраної клітинки встановлюємо величину перевезення (аналогічно методу мінімального елемента);
- викреслюємо рядок і/або стовпець (аналогічно методу мінімального елемента).

Далі цей процес повторюємо з незаповненою частиною таблиці ТЗ. Отриманий план є опорним, оскільки з його клітинок, заповнених додатними перевезеннями, не можна скласти цикл.

Приклад 3.7. Знайти ОП задачі з прикладу 3.4 методом Фоґеля.

$a_i \backslash b_j$	5	15	15	15	Штраф для рядків		
15	8	5 15	13	12	8-5=3	7	–
25/15	6	11	15 10	18 15	11-6=5	4	3
10/5	2 5	9	11 5	16	9-2=7	2	5
Штраф для стовпців	6-2=4	11-9 = 2	13-11 = 2	16-12 = 4			
	–	2	2	4			
	–	–	4	2			

Послідовність заповнення клітинок: (3,1),(1,2),(3,3),(3,2),(2,4). Отриманий опорний план є виродженим ($5 < 3+4-1$); до числа базисних додамо клітинку (1,4) Сумарна вартість перевезення $z = 5 \cdot 2 + 15 \cdot 5 + 10 \cdot 15 + 5 \cdot 11 + 15 \cdot 18 = 560$ ($z = 620$ – у прикладі 3.4). Значення цільової функції таке саме, як і в методі мінімального елемента. Зазвичай, метод Фоґеля дає найкращий початковий розв'язок ТЗ.

3.5. Критерій оптимальності плану перевезень

Для ТЗ існує двоїста до неї задача. Змінні двоїстої задачі позначимо $y_1, y_2, \dots, y_n, -z_1, -z_2, \dots, -z_m$. Задача (3.1) – (3.4) зводиться до такої двоїстої задачі:

$$\sum_{j=1}^n b_j y_j - \sum_{i=1}^m a_i z_i \rightarrow \max; \quad (3.6)$$

$$y_i - z_i \leq c_{ij}, (i = \overline{1, m}; j = \overline{1, n}). \quad (3.7)$$

Основним методом відшукування оптимального плану перевезень є метод потенціалів, в основу якого покладено теорему 3.5.

Теорема 3.5. Для того, щоб деякий план $\bar{X}$ ТЗ був оптимальним, необхідно і достатньо, щоб йому відповідала система із $m+n$ чисел $v_1, v_2, \dots, v_n, u_1, u_2, \dots, u_m$, для якої:

$$v_j - u_i \leq c_{ij}, (i = \overline{1, m}; j = \overline{1, n}); \quad (3.8)$$

$$v_j - u_i = c_{ij} \quad \text{для тих } (i, j), \text{ що } x_{ij} > 0. \quad (3.9)$$

Умови (3.8), (3.9) називають умовами потенціальності, а числа $v_j (j = \overline{1, n})$ та $u_i (i = \overline{1, m})$ – потенціалами відповідних пунктів постачання і споживання. Якщо вважати, що u_i – вартість одиниці продукції у пункті A_i , а v_j – вартість одиниці продукції після перевезення у пункт B_j , то економічний зміст теореми 3.5 такий:

сумарні транспортні витрати за оптимальним планом перевезення дорівнюють збільшенню сумарної вартості продукції після її перевезення у пункти споживання, тобто

$$\sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} = \sum_{j=1}^n b_j v_j - \sum_{i=1}^m a_i u_i. \quad (3.10)$$

3.6. Алгоритм методу потенціалів

Алгоритм методу потенціалів складається з попереднього етапу і основного етапу, який повторюється.

На попередньому етапі необхідно виконати такі дії:

1. Скласти початковий опорний план $\overline{X}_{on}$ ТЗ і визначити множину базисних клітинок B (клітинки, що містять додатні перевезення і ті нульові перевезення, що доповнюють план до невивродженого випадку).
2. Для $\overline{X}_{on}$ скласти систему з $m+n$ чисел $v_j (j = \overline{1, n})$ та $u_i (i = \overline{1, m})$, таких, що виконується умова

$$v_j - u_i = c_{ij}, (i, j) \in B. \quad (3.11)$$

3. План $\overline{X}_{on}$ перевірити на оптимальність, тобто

$$v_j - u_i \leq c_{ij}, (i, j) \notin B.$$

Увага! Система (3.11) має $m+n-1$ рівнянь з $m+n$ невідомими. Для розв'язання системи досить, щоб $u_1 = 0$. Тоді значення інших невідомих можна автоматично, одне за одним, знайти з системи, адже всі базові клітинки з'єднуються ланцюжком.

Основний етап виконується, якщо $\overline{X}_{on}$, знайдений на попередньому етапі, не є оптимальним. Основний етап має три кроки:

1. Поліпшення опорного плану – перехід до нового опорного плану $\overline{X}'_{on}$ такого, що $z(\overline{X}'_{on}) \leq z(\overline{X}_{on})$.
2. Для $\overline{X}'_{on}$ скласти систему з $m+n$ чисел $v_j (j = \overline{1, n})$ та $u_i (i = \overline{1, m})$, таких, що виконується умова

$$v'_j - u'_i = c_{ij}, (i, j) \in B', \quad (3.12)$$

де B' – множина базових клітинок плану $\overline{X'}_{on}$.

3. План $\overline{X'}_{on}$ перевіряється на оптимальність: $v'_j - u'_i \leq c_{ij}, (i, j) \notin B'$.

Основний етап триває доти, поки не знайдено оптимальний план.

Поліпшення опорного плану. Нехай опорний план X не є оптимальним. Тоді існує хоча б одна клітинка (i, j) , для якої виконується нерівність $v'_j - u'_i - c_{ij} > 0$. Серед усіх цих клітинок визначаємо таку клітинку (i_0, j_0) , для якої

$$\max(v_j - u_i - c_{ij}) = v_{j_0} - u_{i_0} - c_{i_0 j_0} > 0. \quad (3.13)$$

Якщо максимум маємо для декількох пар індексів (i, j) , то за (i_0, j_0) вибираємо пару, якій відповідає мінімальна вартість перевезень. Клітинка (i_0, j_0) з частиною базисних клітинок плану, згідно з наслідком 3.2, утворює цикл γ .

Побудуємо цей цикл проти годинникової стрілки, починаючи з клітинки (i_0, j_0) , і позначимо його клітинки поперемінно знаками «+» і «-» (клітинка (i_0, j_0) має знак «+»). Клітинки, позначені знаком «+», утворюють додатний півланцюжок γ^+ , а знаком «-» – від'ємний півланцюжок γ^- . Нехай $\theta = \min_{(i,j) \in \gamma^-} x_{ij} = x_{st}$.

Якщо мінімум досягається для декількох пар індексів (i, j) , то за (s, t) вибираємо ту пару, якій відповідає максимальна вартість перевезень. Новий опорний план будуємо за правилом:

$$x'_{ij} = \begin{cases} x_{ij} - \theta, & \text{якщо } (i, j) \in \gamma^-; \\ x_{ij} + \theta, & \text{якщо } (i, j) \in \gamma^+; \\ x_{ij} & \text{для решти базисних клітинок.} \end{cases} \quad (3.14)$$

Внаслідок (3.14) маємо новий опорний план $\overline{X'}_{on}$, у якому:

- клітинка (s, t) виводиться з базису ($x_{st} = 0$);
- клітинка (i_0, j_0) вводиться у базис ($x_{i_0 j_0} = \theta$).

Залишилося продемонструвати, що при виконанні (3.13) і (3.14) отримано зменшення (принаймні, не збільшення) цільової функції, тобто $z(\overline{X'}_{on}) \leq z(\overline{X}_{on})$.

Нехай I – множина пар індексів (i, j) , які відповідають клітинкам, що не увійшли у замкнутий ланцюжок γ вигляду $(i_0, j_0), (i_0, j_1), (i_1, j_1), (i_1, j_2), \dots, (i_{s-1}, j_{s-1}), (i_{s-1}, j_s), (i_s, j_s), (i_s, j_0)$.

$$\begin{aligned}
\text{Тоді } z\left(\overline{X}'_{on}\right) &= \sum_{i=1}^m \sum_{j=1}^n c_{ij} x'_{ij} = \sum_{(i,j) \in I} c_{ij} x'_{ij} + \sum_{(i,j) \in \gamma^+} c_{ij} x'_{ij} + \sum_{(i,j) \in \gamma^-} c_{ij} x'_{ij} = \\
&= \sum_{(i,j) \in I} c_{ij} x_{ij} + \sum_{(i,j) \in \gamma^+} c_{ij} (x_{ij} + \theta) + \sum_{(i,j) \in \gamma^-} c_{ij} (x_{ij} - \theta) = \\
&= \sum_{(i,j) \in I} c_{ij} x_{ij} + \sum_{(i,j) \in \gamma^+} c_{ij} x_{ij} + \sum_{(i,j) \in \gamma^-} c_{ij} x_{ij} + \theta \left(\sum_{(i,j) \in \gamma^+} c_{ij} - \sum_{(i,j) \in \gamma^-} c_{ij} \right) = \\
&= \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} + \theta (c_{i_0 j_0} - c_{i_0 j_1} + c_{i_1 j_1} - \dots + c_{i_{s-1} j_{s-1}} - c_{i_{s-1} j_s} + c_{i_s j_s} - c_{i_s j_0}) = \\
&= z(\overline{X}_{on}) + \theta (c_{i_0 j_0} - (v_{j_1} - u_{i_0}) + (v_{j_1} - u_{i_1}) - (v_{j_2} - u_{i_1}) - \dots + \\
&+ (v_{j_{s-1}} - u_{i_{s-1}}) - (v_{j_s} - u_{i_{s-1}}) + (v_{j_s} - u_{i_s}) - (v_{j_0} - u_{i_s})) = \\
&= z(\overline{X}_{on}) + \theta (c_{i_0 j_0} - (v_{j_0} - u_{i_0})) = z(\overline{X}_{on}) - \theta (v_{j_0} - u_{i_0} - c_{i_0 j_0}).
\end{aligned}$$

Враховуючи (3.13), отримуємо $z\left(\overline{X}'_{on}\right) \leq z\left(\overline{X}_{on}\right)$.

Приклад 3.8. Знайти розв'язок ТЗ, яку задано опорним планом:

$a_i \backslash b_j$	150	120	80	50
100	100 3	5	7	11
130	50 1	80 4	6	3
170	5	40 8	80 12	50 7

Початковий опорний план не вироджений ($6 = 3 + 4 - 1$), значення лінійної форми $z = 100 \cdot 3 + 50 \cdot 1 + 80 \cdot 4 + 40 \cdot 8 + 80 \cdot 12 + 50 \cdot 7 = 2300$.

Знайдемо потенціали $u_1, u_2, u_3, v_1, v_2, v_3, v_4$ із системи рівнянь:

$$v_1 - u_1 = 3; v_1 - u_2 = 1; v_2 - u_2 = 4; v_2 - u_3 = 8; v_3 - u_3 = 12; v_4 - u_3 = 7.$$

Нехай $u_1 = 0$. Визначимо $v_1 = 3; u_2 = 2; v_2 = 6; u_3 = -2; v_3 = 10; v_4 = 5$.

$a_i \backslash b_j$	3	6	10	5
0	- 100 3	5	+ 7	11
2	+ 50 1	- 80 4	6	3
-2	5	+ 40 8	- 80 12	50 7

Перевіримо опорний план на оптимальність. Серед величин $v_j - u_i - c_{ij}$ для вільних клітинок є додатні:

$$v_2 - u_1 - 5 = 6 - 0 - 5 = 1; v_3 - u_1 - 7 = 10 - 0 - 7 = 3; v_3 - u_2 - 6 = 10 - 2 - 6 = 2.$$

Отже, $(i_0, j_0) = (1, 3)$. У таблиці сформуємо додатний і від'ємний півланцюжки. Знайдемо $\theta = \min_{(i,j) \in \gamma^-} x_{ij} = x_{33} = 80$. У поліпшеному плані замість

клітинки $(3, 3)$ вводимо клітинку $(1, 3)$. Перерахуємо значення потенціалів:

$$v'_1 - u'_1 = 3; v'_1 - u'_2 = 1; v'_2 - u'_2 = 4; v'_2 - u'_3 = 8; v'_3 - u'_1 = 7; v'_4 - u'_3 = 7.$$

Нехай $u'_1 = 0$. Визначимо $v'_1 = 3; u'_2 = 2; v'_2 = 6; u'_3 = -2; v'_3 = 7; v'_4 = 5$.

Поліпшений опорний план з новими значеннями потенціалів:

$a_i \backslash b_j$	3	6	7	5
0	- 3 20	+ 5	80 7	11
2	+ 1 130	- 4 0	6	3
-2	5	8 120	12	7 50

Порушення оптимальності у клітинці $(1, 2)$:

$v_2 - u_1 - 5 = 6 - 0 - 5 = 1 > 0$. Отже, $(i_0, j_0) = (1, 2)$. У таблиці сформуємо додатний та від'ємний півланцюжки. Знайдемо $\theta = \min_{(i,j) \in \gamma^-} x_{ij} = x_{22} = 0$.

У поліпшеному плані замість клітинки $(2, 2)$ вводимо клітинку $(1, 2)$.

Перерахуємо значення потенціалів:

$$v'_1 - u'_1 = 3; v'_1 - u'_2 = 1; v'_2 - u'_1 = 5; v'_2 - u'_3 = 8; v'_3 - u'_1 = 7; v'_4 - u'_3 = 7.$$

Нехай $u'_1 = 0$. Визначимо $v'_1 = 3; u'_2 = 2; v'_2 = 5; u'_3 = -3; v'_3 = 7; v'_4 = 4$.

Поліпшений опорний план з новими значеннями потенціалів:

$a_i \backslash b_j$	3	5	7	4
0	- 3 20	+ 5	80 7	11
2	1 130	4 0	6	3
-3	5 +	8 120	12	7 50

Порушення оптимальності у клітинці $(3, 1)$: $v_1 - u_3 - 5 = 3 + 3 - 5 = 1 > 0$.

Отже, $(i_0, j_0) = (3, 1)$. У таблиці сформуємо додатний та від'ємний півланцюжки. Знайдемо $\theta = \min_{(i,j) \in \gamma^-} x_{ij} = x_{11} = 20$.

У поліпшеному плані замість клітинки $(1, 1)$ вводимо клітинку $(3, 1)$.

Перерахуємо значення потенціалів:

$$v'_1 - u'_2 = 1; v'_1 - u'_3 = 5; v'_2 - u'_1 = 5; v'_2 - u'_3 = 8; v'_3 - u'_1 = 7; v'_4 - u'_3 = 7.$$

Нехай $u'_1 = 0$. Визначимо $v'_2 = 5; u'_3 = -3; v'_4 = 4; v'_1 = 2; u'_2 = 1; v'_3 = 7$.

Поліпшений опорний план з новими значеннями потенціалів

$a_i \backslash b_j$	2	5	7	4
0	3	5 20	7 80	11
1	1 130	4 0	6	3
-3	5 20	8 100	12	7 50

Отриманий план є оптимальним. Значення лінійної форми $z = 130 \cdot 1 + 20 \cdot 5 + 20 \cdot 5 + 100 \cdot 8 + 80 \cdot 7 + 50 \cdot 7 = 2040$. Для знаходження оптимального плану були здійснені три кроки поліпшення опорного плану. Значення функції мети зменшилось на 260 одиниць.

Приклад 3.9. Знайти розв'язок ТЗ прикладу 3.4, опорний план якої знайдено методом мінімального елемента (приклад 3.6).

$a_i \backslash b_j$	0	5	9	12
0	8	5 15	13	12 0
-6	6	11	15 10	18 15
-2	2 5	9	11 5	16

Знайдемо потенціали $u_1, u_2, u_3, v_1, v_2, v_3, v_4$ із системи рівнянь:

$$v_1 - u_3 = 2; v_2 - u_1 = 5; v_3 - u_2 = 15; v_3 - u_3 = 11; v_4 - u_1 = 12; v_4 - u_2 = 18.$$

Нехай $u_1 = 0$. Визначимо $v_1 = 0; v_2 = 5; v_3 = 9; v_4 = 12; u_2 = -6; u_3 = -2$.

Перевіримо опорний план на оптимальність. Серед величин $v_j - u_i - c_{ij}$ для вільних клітинок додатних немає. Отже отриманий план є оптимальним. Значення лінійної форми $z = 5 \cdot 2 + 15 \cdot 5 + 10 \cdot 15 + 5 \cdot 11 + 15 \cdot 18 = 560$. Таким чином, опорний план, який було знайдено методом мінімального елемента, виявився оптимальним.

3.7. Приклади задач, які зводяться до транспортної задачі

Математична структура ТЗ притаманна для великого класу задач лінійного програмування. Реальний зміст таких задач може бути найрізноманітнішим, зовсім не зв'язаним із задачею перевезення вантажів.

Задача про розподіл площ під посіви сільськогосподарських культур

На ділянках D_1, D_2 і D_3 площею S_1, S_2 і S_3 , відповідно, можна вирощувати сільгоспкультури K_1, K_2, K_3 і K_4 . Планове завдання полягає в отриманні врожаю цих культур у кількостях b_1, b_2, b_3 і b_4 тонн, відповідно. Відома матриця $C = \|c_{ij}\|_{i=\overline{1,3}; j=\overline{1,4}}$, елемент c_{ij} якої дорівнює прибутку в одиницях грошей від реалізації культури $K_j (j = \overline{1,4})$, вирощеної на ділянці $D_i (i = \overline{1,3})$. Урожайність культури $K_j (j = \overline{1,4})$ дорівнює P_j ц/га (урожайність від ділянки не залежить). Скласти математичну модель задачі, що даватиме змогу знайти такий план розподілу площ під посіви сільськогосподарських культур, який передбачатиме максимальний прибуток від реалізації урожаю культур.

Позначимо через x_{ij} площу в га, яку необхідно виокремити на ділянці $D_i (i = \overline{1,3})$ для вирощування культури $K_j (j = \overline{1,4})$.

Цільова функція має вигляд $z = \sum_{i=1}^3 \sum_{j=1}^4 (-c_{ij}) x_{ij} \rightarrow \min$.

Умова використання площ усіх ділянок $\sum_{j=1}^4 x_{ij} = S_i, i = \overline{1,3}$.

Нехай $w_j (j = \overline{1,4})$ – загальна площа (в га), необхідна для вирощування культури K_j . З урахуванням планового завдання та урожайності культур маємо $w_j = b_j(m) : \frac{P_j}{10} \left(\frac{m}{\text{га}} \right) = \frac{b_j \cdot 10}{P_j} (\text{га})$.

Умова цілковитого використання площ усіма культурами:

$$\sum_{i=1}^3 x_{ij} = w_j, \text{ де } w_j = \frac{b_j \cdot 10}{P_j}, j = \overline{1,4}.$$

$$\text{Нехай } C = \begin{pmatrix} 20 & 50 & 24 & 10 \\ 25 & 40 & 10 & 20 \\ 30 & 15 & 20 & 15 \end{pmatrix}; S = \begin{pmatrix} 300 \\ 500 \\ 400 \end{pmatrix}; \bar{b}^T = (600; 1500; 225; 1250), \\ \bar{P}^T = (20; 30; 15; 50).$$

Обчислимо $\bar{w}^T = (300; 500; 150; 250)$. Оскільки $\sum_{i=1}^3 S_i = \sum_{j=1}^4 w_j = 1200$, то

ТЗ є закритою, опорний план якої має вигляд:

$S_i \backslash w_j$	300	500	150	250
300	-20	-50 300	-24	-10
500	-25	-40 200	-10 50	-20 250
400	-30 300	-15	-20 100	-15

Початковий опорний план не вироджений ($6 = 3 + 4 - 1$), значення лінійної форми $z = 100 \cdot 3 + 50 \cdot 1 + 80 \cdot 4 + 40 \cdot 8 + 80 \cdot 12 + 50 \cdot 7 = 2300$.

Знайдемо потенціали $u_1, u_2, u_3, v_1, v_2, v_3, v_4$ із системи рівнянь:

$v_2 - u_1 = -50$; $v_2 - u_2 = -40$; $v_3 - u_2 = -10$; $v_4 - u_2 = -20$; $v_1 - u_3 = -30$; $v_3 - u_3 = -20$. Нехай $u_1 = 0$. Визначимо $v_2 = -50$; $u_2 = -10$; $v_3 = -20$; $v_4 = -30$; $u_3 = 0$; $v_1 = -30$.

$S_i \backslash w_j$	-30	-50	-20	-30
0	-20	-50 300	-24	-10
-10	+ -25	-40 200	-10 50	-20 250
0	-30 300	-15	-20 100	-15

Перевіримо опорний план на оптимальність. Серед величин $v_j - u_i - c_{ij}$ для вільних клітинок є додатні: $v_1 - u_2 + 25 = -30 + 10 + 25 = 5$; $v_3 - u_1 + 24 = -20 - 0 + 24 = 4$.

Отже, $(i_0, j_0) = (2, 1)$. У таблиці сформуємо додатний і від'ємний півланцюжки. Знайдемо $\theta = \min_{(i,j) \in \gamma^-} x_{ij} = x_{23} = 50$. У поліпшеному плані замість

клітинки $(2, 3)$ вводимо клітинку $(2, 1)$. Перерахуємо значення потенціалів:

$v'_2 - u'_1 = -50$; $v'_1 - u'_2 = -25$; $v'_2 - u'_2 = -40$; $v'_4 - u'_2 = -20$; $v'_1 - u'_3 = -30$; $v'_3 - u'_3 = -20$.

Нехай $u'_1 = 0$. Визначимо $v'_2 = -50$; $u'_2 = -10$; $v'_1 = -35$; $v'_4 = -30$; $u'_3 = -5$; $v'_3 = -25$.

Поліпшений опорний план з новими значеннями потенціалів:

$S_i \backslash w_j$	-35	-50	-25	-30
0	-20	-50 300	-24	-10
-10	+ -25 50	-40 200	-10	-20 250
-5	-30 250	-15	-20 150	-15

Отриманий план є оптимальним. Значення лінійної форми $z = 50 \cdot (-25) + 250 \cdot (-30) + 300 \cdot (-50) + 200 \cdot (-40) + 150 \cdot (-20) + 250 \cdot (-20) = -39750$. Отже максимальний прибуток від реалізації урожаю культур становить 39750 грошових одиниць.

Задача оптимізації сезонних запасів

Деяка фабрика частину своїх виробничих потужностей використовує для виготовлення сезонних виробів, попит на які підтримується впродовж чотирьох місяців. Замовлення на вироби у ці місяці, відповідно, дорівнюють 100, 200, 180 і 300 одиниць виробів. Протягом цих місяців фабрика може виготовити, відповідно, 50, 180, 280 і 270 одиниць виробів. Витрати на виготовлення одиниці виробу становлять 4 умовні одиниці (у.о.). Зберігання одиниці виробу на складі протягом одного місяця дорівнює 0,5 у.о. Штраф оптовому покупцеві за недопоставку одиниці виробу дорівнює 2 у.о. Скласти план постачання готової продукції оптовому покупцеві, що даватиме змогу мінімізувати витрати, пов'язані з проектом реалізації сезонних виробів протягом чотирьох місяців.

Цю задачу можна сформулювати у термінах ТЗ, якщо:

- пункт постачання $i (i = \overline{1,4})$ замінити місяцем виробництва;
- пункт споживання $j (j = \overline{1,4})$ замінити місяцем постачання;
- величину запасів $a_i (i = \overline{1,4})$ замінити кількістю виготовлених одиниць виробу в i -му місяці;
- величину потреб $b_j (j = \overline{1,4})$ замінити кількістю відпущених одиниць виробу в j -му місяці;
- за c_{ij} вважати витрати, що стосуються постачання у j -му місяці $(j = \overline{1,4})$, одиниці виробу, виробленої в i -му місяці $(i = \overline{1,4})$, які визначаються формулою

$$c_{ij} = \begin{cases} 4, & \text{якщо } i = j; \\ 4 + 0,5(j - i), & \text{якщо } i < j; \\ 4 + 2(i - j), & \text{якщо } i > j. \end{cases}$$

Дві інші класичні задачі дослідження операцій (задачу про призначення і ТЗ за критерієм часу) розглянемо в окремих параграфах, оскільки для їхнього розв'язання розроблені спеціальні методи.

3.8. Транспортна задача за критерієм часу

У деяких ТЗ за критерій беруть час t_{ij} перевезення однорідних вантажів від i -го пункту постачання ($i = \overline{1, m}$) до j -го ($j = \overline{1, n}$) пункту споживання (перевезення свіжих продуктів, евакуація людей тощо). Задача полягає у визначенні такого плану перевезень $\|x_{ij}\|_{mn}$, за яким весь вантаж буде доставлено споживачам у найкоротший термін згідно з умовами (3.2) – (3.5).

Розглянемо постановку задачі детальніше. Серед значень t_{ij} виокремимо найбільше (позначимо $t_{\max}$). Величина $t_{\max}$ визначає час, протягом якого виконається поточний план перевезень $\|x_{ij}\|_{mn}$. Необхідно визначити такий план, щоб $t_{\max}$ було найменшим.

Початковий опорний план ТЗ знаходимо одним з відомих методів (північно-західного кута, мінімального елемента тощо). Нехай клітинка (k, l) цього плану містить додатне перевезення, що вимагає найбільшого часу (позначимо $t_{kl} = T$).

Починаючи з опорного плану, будуватимемо наступні плани, визначаючи на кожній ітерації досягнуте значення T . Усі вільні клітинки, яким відповідають значення $t_{ij} > T$, завантажувати недоцільно, оскільки це збільшить T . Отож, викреслюючи ці клітинки, вилучаємо їх з подальшого розгляду.

У таблиці, що залишилася, треба вилучити клітинку (k, l) . З цією метою побудуємо розвантажувальний цикл, який складатиметься з базисних і вільних комірок. Від'ємний півланцюжок γ^- починається з клітинки (k, l) . Решта клітинок цього півланцюжка – базисні клітинки (i, j) . У додатний півланцюжок γ^+ можуть входити як базисні, так і вільні клітинки. Якщо ланцюжок γ вдалося побудувати, то застосовуємо формулу (3.14).

Якщо $\theta = x_{kl}$, то клітинка (k, l) цілковито звільняється і в подальшому не розглядається (закреслюється). Якщо ж $\theta < x_{kl}$, то вантаж у цій клітинці зменшується: $x'_{kl} = x_{kl} - \theta$. У цьому випадку будуватимемо новий розвантажувальний цикл доти, поки не отримаємо $x'_{kl} = 0$. На цьому одна ітерація закінчується. Далі знову визначатимемо нове значення $T' < T$, закреслюючи клітинки зі значеннями $t_{ij} > T'$. Продовжуватимемо цей процес доти, поки на деякій ітерації вже неможливо буде побудувати ланцюжок γ . Це означатиме, що досягнуто оптимального значення.

Приклад 3.12. Розв'язати транспортну задачу за критерієм часу, знаючи її опорний план. Домовимося значення $t_{\max}$ підкреслювати.

b_j	5	15	15	15
a_i				
15	21	- 2 15	20	+ 11 0
25	12	7	9	- 20 10
10	4	14	16	18

Ланцюжок γ вдалося побудувати. Застосовуємо формулу (3.14).

b_j	5	15	15	15
a_i				
15		- 2 5		+ 11 10
25	12	+ 7 10	- 9 15	20
10	4	14	+ 16	- 18 5

Ланцюжок γ вдалося побудувати. Застосовуємо формулу (3.14).

b_j	5	15	15	15
a_i				
15		2		11
25	12	+ 7 10	- 9 10	20
10	4	+ 14	- 16 5	18

Ланцюжок γ вдалося побудувати. Застосовуємо формулу (3.14).

b_j	5	15	15	15
a_i				
15		2		11
25	12	7	9	
10	4	14	16	

Ланцюжок γ вдалося побудувати. План оптимальний.

Бібліографічний список

1. Вагнер Г. Основы исследования операций в 2 т. / Г. Вагнер. – М.: Мир, 1973. – 824 с.
2. Корбут, А. А. Дискретное программирование / А. А. Корбут, Ю. Ю. Финкельштейн. – М.: Наука, 1969. – 368 с.
3. Гончаренко Я. В. Математичне програмування / Я. В. Гончаренко. – Київ: НПУ ім. М. П. Драгоманова, 2010. – 184 с.
4. Бартіш, М. Я. Дослідження операцій / Підручник / М. Я. Бартіш, І. М. Дудзяний. – Львів: Видавничий центр ЛНУ ім. Івана Франка, 2007. – 168 с.
5. Оптимізаційні методи і моделі в економіці і менеджменті. Текст лекцій з курсу «Економіко-математичні методи та моделі» / О. Є. Скворчевський. – Харків : НТУ «ХПІ», 2014. – 76 с.
6. Решение задачи коммивояжера методом целочисленного линейного программирования с последовательным исключением подциклов: описание и алгоритмическая реализация / М. В. Козлов, Ф. В. Костюк, С. В. Сорокин, А. В. Тюленев. – Advanced Science. 2012. № 2. С. 124 – 141.
7. Сигал, И. Х. Введение в прикладное дискретное программирование: модели и вычислительные алгоритмы / И. Х. Сигал, А. П. Иванова. – М.: ФИЗМАТЛИТ, 2003. – 240 с.
8. Вітлінський, В. В. Математичне програмування / В. В. Вітлінський, С. І. Наконечний, Т. О. Терещенко. – Київ: КНЕУ, 2001. – 248 с.
9. Моклячук, М. П. Теорія вибору та прийняття рішень / М. П. Моклячук, Р. Є. Ямненко. – Київ: ВПЦ «Київськ. ун-т.», 2013. – 528 с.
10. Барвінський А. Ф. Математичне програмування: навч. посіб. / А. Ф. Барвінський – Львів: Нац. ун-т “Львівська політехніка”, 2004. – 448 с.
11. Демиденко, М. А. Математичне програмування / М. А. Демиденко. – Дніпропетровськ, НГУ, 2005. – 114 с.
12. Щуровська А. Ю. Математичне програмування: навч. посіб. / А. Ю. Щуровська. – Одеса, 2008. – 139 с.

Навчальне видання

**Барахов Костянтин Петрович
Курєннов Сергій Сергійович**

МЕТОДИ ЛІНІЙНОЇ ТА ЦІЛОЧИСЛОВОЇ ОПТИМІЗАЦІЇ

Редактор Є.О. Александрова

Зв. план, 2019

Підписано до видання 22.03.2019

Ум. друк. арк. 3,6. Обл.-вид. арк. 4. Електронний ресурс

Видавець і виготовлювач
Національний аерокосмічний університет ім. М. Є. Жуковського
«Харківський авіаційний інститут»
61070, Харків-70, вул. Чкалова, 17
<http://www.khai.edu>
Видавничий центр «ХАІ»
61070, Харків-70, вул. Чкалова, 17
izdat@khai.edu

Свідоцтво про внесення суб'єкта видавничої справи
до Державного реєстру видавців, виготовлювачів і розповсюджувачів
видавничої продукції сер. ДК № 391 від 30.03.2001