ДОСЛІДЖЕННЯ ОПЕРАЦІЙ

МОДЕЛІ ТА ЗАДАЧІ

В.М.Кирилич, В. А. Козицький

Тексти лекцій

Львів - 2012

УДК 519.85(075.8)
ББК В 173 я73
К-43

Рецензенти:

д-р. фіз.-мат. наук, проф. М. Я. Бартіш
(Львівський національний університет імені Івана Франка);
д-р. фіз.-мат. наук, проф. М.М.Притула
(Львівський національний університет імені Івана Франка).

Рекомендовано до друку Вченою радою механіко-математичного факультету Львівського національного університету імені Івана Франка
Протокол № 9 від 06.06. 2012 р.

Кирилич В.М.

К- 43 **Дослідження операцій. Моделі та задачі**: тексти лекцій/ В.М.Кирилич, В.А. Козицький. – Львів: ЛНУ імені Івана Франка, 2012. – 140с.
.

Розглянуто задачі лінійного, цілочислового і динамічного програмування та мережеві задачі. Наведено застосування дослідження операцій до задач економіки.

Для спеціалістів, магістрів економічних і математичних спеціальностей.

УДК 519.85(075.8)
ББК В 173 я73

© Кирилич В.М., Козицький В.А., 2012
© Львівський національний університет імені Івана Франка, 2012

Навчальне видання

Кирилич Володимир Михайлович,
Козицький Валерій Андрійович

Дослідження операцій. Моделі та задачі

Тексти лекцій

Редактор *Н. Й. Плиса*
Технічний редактор *С. З. Сеник*
Комп'ютерний набір і верстання *В. А. Козицький*
Обкладинка *Василь Роган*

Формат 60×90/16. Умовн. друк. арк. 8,6. Тираж 100 прим.

Видавець і виготовляч:
Львівський національний університет імені Івана Франка,

вул. Університетська, 1, м. Львів, 79000.

chapter]
[chapter] [chapter] [chapter]
[chapter]

Зміст

Вступ **5**

1 Задача лінійного програмування — LP **7**
1.1 Фундаментальні теореми лінійного програмування 9
1.2 Базисні розв'язки 16
1.3 LP і симплекс-метод 19
1.4 Аналіз чутливості 27
1.5 LP з двосторонніми обмеженнями 32
1.6 Транспортна задача 38

2 Цілочислове програмування **53**
2.1 Формулювання задачі цілочислового LP 53
2.2 Метод Гоморі 54
2.3 Метод гілок і меж 66
2.4 Задача булевого програмування 74

3 Динамічне програмування **82**
3.1 Метод динамічного програмування 82
3.2 Рівняння Белмана 85
3.3 Задача інвестування 93
3.4 Задача про заміну обладнання 97
3.5 Задача планування виробництва та запасів 101
3.6 Задача про найкоротший шлях на мережі 105

4 Потоки в мережі **111**
4.1 Задача про найкоротший шлях 111
4.2 Мережеве планування 117
4.3 Задача про максимальний потік 125
4.4 Пошук максимального потоку 128

Список літератури **136**

Вступ

Операція — це система керованих дій, які об'єднані єдиним задумом і спрямовані на досягнення визначеної цілі. Наприклад, треба організувати побудову аеропорту і зазначити порядок виконання робіт у часі і розподілити необхідні ресурси між видами робіт так, щоб закінчити будівництво об'єкта в заданий час і з найменшою вартістю.

Дослідження операцій — це комплексна математична дисципліна, яка займається побудовою, аналізом і застосуванням математичних моделей прийняття оптимальних рішень при проведені операції.

Набір параметрів керування (змінних) при проведені операції називається розв'язком. Розв'язок, який задовольняє набір деяких умов, називається допустимим. Допустимий розв'язок, який за деяких визначених ознак має перевагу над іншими або принаймні не гірший за них, називається оптимальним. Ознака переваги є критерієм оптимальності. Критерій оптимальності — це цільова функція і напрям оптимізації або набір цільових функцій і відповідних напрямів оптимізації. Цільова функція визначає ефективність розв'язків, а напрям оптимізації — це максимум (мінімум), якщо найліпшим є найбільше (найменше) значення цільової функції. Наприклад, критерієм може бути максимізація прибутку або мінімізація вартості.

Математична модель дослідження операцій складається: з опису параметрів керування (змінних), які потрібно знайти, опису критерію оптимальності й опису множини допустимих розв'язків. Задачі дослідження операцій класифікують залежно від параметрів задачі від часу: статичні задачі (прийняття рішення відбувається за умов, коли всі параметри задач наперед відомі і не змінюються з часом) та динамічні задачі (в процесі прийняття рішення параметри задачі змінюються з часом і процедура прийняття рішення виконується кроками та може бути подана дискретним або неперервним процесом). Класифікують задачі дослідження операцій також за достовірністю інформації (детерміновані та недетерміновані задачі) і за критерієм оптимальності

(критерії оптимальності можуть мати довільний вигляд, зокрема неформалізований). Найпоширеніший (формалізований) критерій оптимальності у вигляді оптимізації (максимізації або мінімізації) однієї або декількох скалярних цільових функцій. До однокретирійних задач оптимізації — оптимізації скалярної функції на заданій множині допустимих числових розв'язків (задач математичного програмування) зачисляють наприклад, задачу лінійного програмування, задачу цілочислового програмування, задачу булевого програмування, задачу динамічного програмування та ін. У задачах дослідження операцій в більшості випадків наявний не один, а декілька критеріїв оцінки. Такі задачі називають багатокритерійними.

Для написання текстів лекцій використали підручники [4,9,10], в яких розглянуто теми, які мало або зовсім не відображені в дисципліні "Методи оптимізації та варіаційне числення", яку читають на механіко-математичному факультеті, а саме: двоїстий симплекс-метод, задача лінійного програмування з двосторонніми обмеженнями, задача цілочислового програмування, метод динамічного програмування і мережеві задачі.

Розділ 1

Задача лінійного програмування — LP

Задачею лінійного програмування — LP називається задача максимізації (або мінімізації) афінної функції при лінійних обмеженнях: рівностях і (або) нерівностях, тобто оптимізація афінної функції на поліедрі

$$\begin{aligned} &\max\ c\cdot x\ + d, \\ \text{за умов}\quad &Gx \leqslant q, \\ &Ax = b, \quad x \in \mathbb{R}^n, \end{aligned} \tag{1.1}$$

де $G \in \mathbb{R}^{k\times n},\ A \in \mathbb{R}^{m\times n},\ c \in \mathbb{R}^n,\ q \in \mathbb{R}^k,\ b \in \mathbb{R}^m,\ d \in \mathbb{R}$.

Отож, LP — задача максимізації афінної функції $c\cdot x+d$ (або еквівалентно, лінійної функції $c\cdot x$) на поліедрі $\mathcal{P} = \{x \in \mathbb{R}^n : Gx \leqslant q,\ Ax = b\}$.

Прийнято декілька форм запису (LP), кожна з яких зручна, коли розглядають певну задачу. Задача (1.1) — LP в загальній формі. *Канонічна форма* LP, коли обмеження-нерівності мають лише вигляд $x \geqslant 0$, тобто

$$\begin{aligned} &\max\ c\cdot x, \\ \text{за умов}\quad &Ax\ = b, \\ &x \geqslant 0. \end{aligned} \tag{1.2}$$

Стандартна форма LP, коли немає обмеження-рівності

$$\begin{aligned} &\max\ c\cdot x,\\ &\text{за умов}\quad Ax \leqslant b. \end{aligned} \tag{1.3}$$

Задачу LP у загальній формі можна записати або в канонічній формі, або в стандартній формі. Отож, задача LP набуде канонічної форми, якщо введемо додаткові змінні s_i і зробимо заміну змінних $x = x^+ - x^-$, x^+, $x^- \geqslant 0$. Тоді задача набуде вигляду

$$\begin{aligned} &\max\ c\cdot x^+ - c\cdot x^- + d,\\ &\text{за умов}\quad Gx^+ - Gx^- + s = q,\\ &\qquad\quad Ax^+ - Ax^- = b,\\ &\qquad\quad x^+ \geqslant 0,\ x^- \geqslant 0,\ \ s \geqslant 0. \end{aligned} \tag{1.4}$$

Задача лінійного програмування в канонічній формі зі змінними x^+, x^- та s.

Багато важливих задач економіки та організації виробництва можуть бути зведені до задачі лінійного програмування без втрати загальності.

Приклад 1.1. Задача про дієту. Історично цю задачу лінійного програмування розглянули одну з перших. Її сформулювали та розв'язали в цінах 1944 р. Зокрема, для оплати відповідної дієти треба приблизно 60 доларів на рік, якщо ця дієта відповідає смакам тих рідкісних споживачів, які можуть отримувати задоволення від одноманітної їжі.

Задача лінійного програмування, яка нагадує задачу про дієту, виникає в нафтовій промисловості, наприклад, під час розрахунку найдешевшого способу змішування деяких сортів і кількості бензину для отримання декількох видів палива. Кожний вид палива повинен мати визначену кількість властивостей.

Нехай є n продуктів споживання і m корисних елементів. Відомі такі параметри:

- a_{ij} — вміст i-го елемента в одиниці j-го продукту;
- c_j — ціна за одиницю j-го продукту;

- b_j — потреба споживача в i-му елементі.

Позначимо через x_j споживання j-го продукту споживачем. Тоді задача про дієту набуде вигляду

$$\begin{aligned} &\min\ c\cdot x,\\ &\text{за умов}\quad Ax\geqslant b,\quad x\geqslant 0. \end{aligned}$$

▲

Приклад 1.2. Модель аналізу технологічних процесів. Нехай в економічній системі є n чинників виробництва і m — технологічних процесів. Нехай a_{ij} — кількість i-го чинника, яка потрібна для функціонування з одиничною інтенсивністю j-го технологічного процесу. Інтенсивність процесу може визначатися кількістю фірм, які використовують цей процес, і рівнем затрат трудових ресурсів, які розглядають як екзогенний чинник.

Нехай b_i — наявна кількість i-го чинника, c_j — ціна продукції, яку отримують внаслідок використання j-го процесу з одиничною інтенсивністю, x_j — шукана інтенсивність j-го процесу.

Отже, задача знаходження вектора інтенсивностей $x\geqslant 0$ за наявних початкових запасів факторів виробництва b шляхом максимізації загальної вартості виробленої продукції набуде вигляду

$$\begin{aligned} &\max\ c\cdot x,\\ &\text{за умов}\quad Ax\leqslant b,\quad x\geqslant 0. \end{aligned}$$

▲

1.1 Фундаментальні теореми лінійного програмування

Опуклий поліедр

Означення 1.1. *Непорожня множина $\mathcal{P}\subseteq\mathbb{R}^n$ називається опуклим поліедром, якщо $\mathcal{P}$ — перетин скінченної кількості замкнених півпросторів, тобто*

$$\mathcal{P}=\{x\in\mathbb{R}^n:\ a_j\cdot x\leqslant\alpha_j,\ (j=1,\dots,m)\}=\bigcap_{j=1}^{m}\mathrm{H}^-_{a_j,\alpha_j},$$

де $\{a_j\}_{j=1}^m$ *— задані вектори з* $\mathbb{R}^n$, $\{\alpha_j\}_{j=1}^m$ *— задані числа.*

Поліедир $\mathcal{P}$ називається (опуклим) *політопом*, якщо $\mathcal{P}$ — обмежена множина.

Наступна теорема є фундаментальною теоремою теорії опуклих поліедрів, вона визначає зображення опуклого поліедра в термінах вершин і напрямів.

Теорема 1.1. (***Мінковського-Вейля***). *Непорожня множина* $\mathcal{P} \subseteq \mathbb{R}^n$ *є опуклим поліедром тоді і лише тоді, коли існують вектори* $\{v_i\}_{i=1}^k$ *і* $\{d_j\}_{j=1}^l$ *такі, що*

$$\begin{aligned}\mathcal{P} &= \mathbf{conv}\{v_1,\dots,v_k\} + \overline{\mathbf{cone}}\{d_1,\dots,d_l\}\\ &= \Big\{\sum_{i=1}^k \lambda_i v_i + \sum_{j=1}^l \mu_j d_j : \sum_{i=1}^k \lambda_i = 1,\ \lambda_i \geqslant 0,\ \mu_j \geqslant 0\Big\}.\end{aligned} \tag{1.5}$$

Наступна афінна версія леми Фаркаша за змістом еквівалентна теоремі сильної двоїстості лінійного програмування.

Лема 1.1. (***Лема Фаркаша, афінна версія***). *Нехай* $\{a_j\}_{j=1}^m$, $a_j\mathbb{R}^n$, $\{\alpha_j\}_{j=1}^m$, $\alpha_j \in \mathbb{R}$, *— задані вектори і числа. Припустимо, що лінійні нерівності*

$$a_j \cdot x \leqslant \alpha_j \ (j = 1,\dots,m)$$

сумісні. Тоді такі твердження еквівалентні:

(i) $\quad [a_j \cdot x \leqslant \alpha_j\,(j = 1,\dots,m)\,] \implies [c \cdot x \leqslant \gamma]$,

(ii) $\quad \exists (y_1,\dots,y_m) \geqslant 0$ *таке, що* $\sum\limits_{i=1}^m a_i y_i = c,\ \sum\limits_{i=1}^m y_i \alpha_i \leqslant \gamma.$

Фундаментальні теореми LP

Розглянемо задачу лінійного програмування

$$\begin{aligned}(\mathrm{P}):\qquad &\max\ \ c \cdot x,\\ &\text{за умов}\quad a_j \cdot x \leqslant b_j\ \ (j = 1,\dots,m),\\ &\qquad\qquad x \in \mathbb{R}^n,\end{aligned} \tag{1.6}$$

до якої можна звести будь-яку іншу задачу LP.

Лема 1.2. *Нехай допустима множина $\mathcal{P}$ задачі (1.6) непорожня. Задача (1.6) має розв'язок тоді і лише тоді, коли цільова функція $c \cdot x$ обмежена зверху на $\mathcal{P}$.*

Доведення. Нехай цільова функція обмежена зверху сталою $M < \infty$ на допустимій множині $\mathcal{P} = \{x \in \mathbb{R}^n : a_j \cdot x \leqslant b_j \ (j = 1, \dots, m)\}$. За теоремою 1.1 $\mathcal{P}$ має зображення

$$\mathcal{P} = \mathbf{conv}\{v_1, \dots, v_k\} + \overline{\mathbf{cone}}\{d_1, \dots, d_l\}.$$

Оскільки $v_1 + td_j \in \mathcal{P}$ для всіх $t \geqslant 0$, то отримаємо $c \cdot (v_1 + td_j) \leqslant M$. Звідси при $t \to \infty$ одержимо $c \cdot d_j \leqslant 0$ для $j = 1, \dots, l$. Тепер супремум цільової функції на $\mathcal{P}$ дорівнює

$$\sup\Big\{\sum_{i=1}^{k} \lambda_i (c \cdot v_i) + \sum_{j=1}^{l} \delta_j (c \cdot d_j) : \lambda \in P^{m-1},\ \delta_j \geqslant 0 \ (j = 1, \dots, l)\Big\}$$

$$= \sup\Big\{\sum_{i=1}^{k} \lambda_i (c \cdot v_i) : \lambda \in P^{m-1}\Big\} = \max\{c \cdot v_i : \ (i = 1, \dots, m)\},$$

де P^{m-1} — стандартний ймовірнісний симплекс в $\mathbb{R}^m$. Ми показали, що супремум цільової функції можна досягнути і досягнути в деякій екстремальній точці v_i.

□

Наступна теорема про існування розв'язку в двоїстій теорії задачі лінійного програмування.

Теорема 1.2. *Нехай задача (1.6) має розв'язок. Допустима точка x^* є розв'язком (1.6) тоді і лише тоді, коли існують множники $\{\lambda_j^*\}_{j=1}^m$ такі, що*

$$\sum_{j=1}^{m} \lambda_j^* a_j = c, \quad \lambda_j^* \geqslant 0, \quad i \quad b \cdot \lambda^* = c \cdot x^*. \tag{1.7}$$

Доведення. Нехай x^* і λ^* задовольняють (1.7). Якщо x допустима точка, то

$$c \cdot x = \Big(\sum_{j=1}^{m} \lambda_j^* a_j\Big) \cdot x = \sum_{j=1}^{m} \lambda_j^* (a_j \cdot x) \leqslant \sum_{j=1}^{m} \lambda_j^* b_j = c \cdot x^*,$$

тобто x^* є розв'язком (1.6).

Навпаки, нехай x^* — розв'язок (1.6). Оскільки кожна допустима точка x задовольняє умову $c\cdot x \leqslant c\cdot x^*$, тобто

$$a_j\cdot x \leqslant b_j\ (j=1,\dots,m)\implies c\cdot x\leqslant c\cdot x^*,$$

то за лемою Фаркаша 1.1 існують невід'ємні $\{\lambda_j^*\}_{j=1}^m$ такі, що $\sum\limits_{j=1}^m \lambda_j^* a_j = c$ і $b\cdot\lambda^*\leqslant c\cdot x^*$. Зокрема,

$$c\cdot x^* - b\cdot\lambda^* = \sum_{j=1}^m \lambda_j^*(a_j\cdot x^*) - b\cdot\lambda^* = \sum_{j=1}^m \lambda_j^*(a_j\cdot x^* - b_j)\leqslant 0,$$

оскільки $\lambda^*\geqslant 0$ і $a_j\cdot x^*\leqslant b_j$. Звідси отримаємо $c\cdot x^*\leqslant b\cdot\lambda^*$, і отже, $c\cdot x^* = b\cdot\lambda^*$. □

Зауваження 1.1. Множники $\{\lambda_j^*\}_1^m$ "засвідчують оптимальність" для x^*: щоб переконатися, що допустима точка x^* справді є оптимальним розв'язком (1.6), потрібно перевірити, що точка x^* є допустимою для задачі і λ^* задовольняє (1.7).

Наслідок 1.3. *Нехай задача (1.6) має розв'язок. Допустима точка x^* є розв'язком (1.6) тоді і лише тоді, коли існують множники $\{\lambda_j^*\}_1^m$ такі, що справджуються умови*

$$\sum_{j=1}^m \lambda_j^* a_j = c,\quad \lambda_j^*\geqslant 0\ (j=1,\dots,m),$$

і ***умови доповняльності****: для кожного $j=1,\dots,m$,*

$$\text{або } \lambda_j^* = 0,\ \text{ або } a_j\cdot x^* = b_j. \tag{1.8}$$

Це випливає з теореми, оскільки $c\cdot x^* = b\cdot\lambda^*$, що еквівалентно

$$b\cdot\lambda^* - c\cdot x^* = \sum_{j=1}^m \lambda_j^*(a_j\cdot x^* - b_j) = 0,$$

позаяк $\lambda_j^*\geqslant 0$ і $a_j\cdot x^* - b_j\leqslant 0$, тому одержуємо (1.8).

Умови доповняльності ще називають умовами доповняльної нежорсткості.

Умова оптимальності

$$c \cdot x \leqslant c \cdot x^* = b \cdot \lambda^* \quad \text{для всіх} \quad x \in \mathcal{P},$$

призводить до розгляду двоїстої задачі лінійного програмування

$$\begin{aligned} (\mathrm{D}): \qquad & \min b \cdot \lambda, \\ & \text{за умов} \quad \sum_{j=1}^{m} \lambda_j a_j = c, \\ & \lambda_j \geqslant 0 \ \ (j = 1, \ldots, m). \end{aligned} \tag{1.9}$$

Стосовно пари задач (1.6) і (1.9) отримали такий фундаментальний результат.

Теорема 1.4. (***Теорема слабкої двоїстості*** LP). *Якщо x — допустимий розв'язок (1.6) і λ — допустимий розв'язок (1.9), то*

$$c \cdot x \leqslant b \cdot \lambda.$$

Доведення. Оскільки $\lambda_j \geqslant 0$ і $a_j \cdot x \leqslant b_j$, то

$$b \cdot \lambda - c \cdot x = b \cdot \lambda - \sum_{j=1}^{m} \lambda_j (a_j \cdot x) = \sum_{j=1}^{m} \lambda_j (b_j - (a_j \cdot x)) \geqslant 0.$$

□

Вираз

$$b \cdot \lambda - c \cdot x$$

називається двоїстим надлишком. Теорема слабкої двоїстості стверджує таке: якщо x і λ допустимі, то двоїстий надлишок невід'ємний.

Теорема 1.5. (***Теорема сильної двоїстості*** LP). *Якщо задача (1.6) має оптимальний розв'язок x^*, тоді (1.9) також має оптимальний розв'язок λ^* і оптимальне значення цих задач однакове, тобто*

$$c \cdot x^* = b \cdot \lambda^*.$$

Доведення теореми випливає з теореми 1.2.

Задачі лінійного програмування (P) і (D) можна записати в компактній формі, використавши матриці. Позначимо через A — $m \times n$ матрицю з рядками a_i^T, тобто $A^T = [a_1, \ldots, a_m]$. Тоді отримаємо

$$\text{(P)}: \quad \begin{aligned} &\max c \cdot x, \\ \text{за умов} \quad &Ax \leqslant b, \\ &x \in \mathbb{R}^n, \end{aligned} \qquad \text{(D)}: \quad \begin{aligned} &\min \quad b \cdot \lambda, \\ \text{за умов} \quad &A^T\lambda = c, \\ &\lambda \geqslant 0. \end{aligned}$$

Правило двоїстості в лінійному програмуванні

Пряма задача (LP)	Двоїста задача (LP)
Максимізація	Мінімізація
i-те обмеження: $\leqslant$ i-те обмеження: $=$ i-те обмеження: $\geqslant$	i-та змінна: $\geqslant 0$ i-та змінна: без обмеження i-та змінна: $\leqslant 0$
j-та змінна: $\geqslant 0$ j-та змінна: без обмеження j-та змінна: $\leqslant 0$	j-те обмеження: $\geqslant$ j-те обмеження: $=$ j-те обмеження: $\leqslant$

Якщо в задачі на максимум обмеження набувають вигляду $\leqslant$ і незалежні зміні невід'ємні, то в двоїстій задачі на мінімум обмеження набувають вигляду $\geqslant$ і незалежні змінні є невід'ємними, тобто пара двоїстих задач набуде вигляду

$$\text{(P)}: \quad \begin{aligned} &\max c \cdot x, \\ \text{за умов} \quad &Ax \leqslant b, \\ &x \geqslant 0, \end{aligned} \qquad \text{(D)}: \quad \begin{aligned} &\min \quad b \cdot \lambda, \\ \text{за умов} \quad &A^T\lambda \geqslant c, \\ &\lambda \geqslant 0, \end{aligned}$$

де A — $m \times n$ матриця, $c, x \in \mathbb{R}^n$ і $b, \lambda \in \mathbb{R}^m$. Наведена пара двоїстих задач лінійного програмування є в симетричній формі. Використовуючи наведене правило, можна показати, що наступні задачі LP є також парою двоїстих задач

$$\text{(P)}: \quad \begin{aligned} &\min c \cdot x, \\ \text{за умов} \quad &Ax = b, \\ &x \geqslant 0, \end{aligned} \qquad \text{(D)}: \quad \begin{aligned} &\max \ b \cdot \lambda, \\ \text{за умов} \quad &A^T\lambda \leqslant c \\ &\lambda \in \mathbb{R}^m. \end{aligned}$$

Кажуть, що така пара двоїстих задач записана в стандартній формі. Симплекс-метод застосовують до задачі лінійного програмування (P) в канонічній формі. Якщо задача лінійного програмування не є записана в канонічній формі, то введенням слабких змінних можна цю задачу звести до канонічної форми.

Наслідок 1.6. (***Умови доповняльності***). *Нехай x^* і λ^* — допустимі розв'язки пари двоїстих задач у симетричній формі. Для того, щоб x^* і λ^* були оптимальними розв'язками, необхідно і достатньо, щоб справджувалися рівності*

$$\begin{aligned} &\text{(i)} \qquad x_j^*\Big(\sum_{i=1}^{m} a_{ij}\lambda_i^* - c_j\Big) = 0, \;\; j = 1, \dots, n, \\ &\text{(ii)} \qquad \lambda_i^*\Big(b_i - \sum_{j=1}^{n} a_{ij}x_j^*\Big) = 0, \;\; i = 1, \dots, m. \end{aligned} \tag{1.10}$$

Приклад 1.3. 1. Розглянемо LP

$$\begin{aligned} (\mathrm{P}): \qquad & \max\, 4x_1 + x_2 + 5x_3 + 3x_4, \\ \text{за умов} \quad & x_1 - x_2 - x_3 + 3x_4 \leqslant 1, \\ & 5x_1 + x_2 + 3x_3 + 8x_4 \geqslant 15, \\ & -x_1 + 2x_2 + 3x_3 - 5x_4 = 3, \\ & x_1 \geqslant 0,\ x_2 \ - \text{вільне},\ x_3 \geqslant 0,\ x_4 \leqslant 0. \end{aligned}$$

Тоді двоїста задача набуде вигляду

$$\begin{aligned} (\mathrm{D}): \qquad & \min\, \lambda_1 + 15\lambda_2 + 3\lambda_3, \\ \text{за умов} \quad & \lambda_1 + 5\lambda_2 - \lambda_3 \geqslant 4, \\ & -\lambda_1 + \lambda_2 + 2\lambda_3 = 1, \\ & -\lambda_1 + 3\lambda_2 + 3\lambda_3 \geqslant 5, \\ & 3\lambda_1 + 8\lambda_2 - 5\lambda_3 \leqslant 3, \\ & \lambda_1 \geqslant 0,\ \lambda_2 \leqslant 0,\ \lambda_3 \ - \text{вільне}. \end{aligned}$$

2. Розглянемо LP

$$\begin{aligned}(\mathrm{P}): \qquad & \min\ 3x_1 - 6x_2 - x_3 + 2x_4,\\ \text{за умов}\quad & 2x_1 - 16x_2 + 12x_3 + 0x_4 \geqslant 12,\\ & 5x_1 + 5x_2 + 0x_3 - 13x_4 \leqslant 23,\\ & -4x_1 - 3x_2 - x_3 - 7x_4 = 28,\\ & x_1 \geqslant 0,\ x_2, x_3 \ -\ \text{вільні},\ x_4 \geqslant 0.\end{aligned}$$

Тоді двоїста задача набуде вигляду

$$\begin{aligned}(\mathrm{D}): \qquad & \max\ 12\lambda_1 - 23\lambda_2 + 28\lambda_3,\\ \text{за умов}\quad & 2\lambda_1 - 5\lambda_2 - 4\lambda_3 \leqslant 3,\\ & -16\lambda_1 - 5\lambda_2 - 3\lambda_3 = -6,\\ & 12\lambda_1 + 0\lambda_2 - \lambda_3 = -1,\\ & 0\lambda_1 + 13\lambda_2 - 7\lambda_3 \leqslant 2,\\ & \lambda_1 \geqslant 0,\ \lambda_2 \geqslant 0,\ \lambda_3 \ -\ \text{вільне}.\end{aligned}$$

▲

1.2 Базисні розв'язки

Розглянемо систему рівнянь

$$Ax = b, \tag{1.11}$$

де $x \in \mathbb{R}^n$, $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$ і $b \geqslant 0$. Нехай $\operatorname{rank} A = \operatorname{rank}(A|b) = m$ і, необмежуючи загальності, припустимо, що перші m стовпців матриці A лінійно незалежні. Позначимо $m \times m$ матрицю складену з цих стовпців через B. Матриця B невироджена і система $Bx_B = b$ має єдиний розв'язок x_B. Приймемо

$$\overline{x} = (x_B, 0). \tag{1.12}$$

Тоді $\overline{x}$ є розв'язком системи $Ax = b$.

Компоненти x, які відповідають стовпцям B, називаються *базисними змінними*, а розв'язок системи (1.11), який набув вигляду (1.12), називається *базисним розв'язком*. Якщо одна або більше базисних змінних у базисному розв'язку набуває значення нуль, то розв'язок називається *виродженим базисним розв'язком.*

Припустимо, що $m \times n$ матриця A, $m < n$, має m лінійно незалежних рядків.

Розглянемо систему обмежень

$$\begin{aligned} Ax &= b, \\ x &\geqslant 0, \end{aligned} \tag{1.13}$$

яка зображає обмеження задачі LP в канонічній формі.

Означення 1.2. *Вектор x, який задовольняє (1.13), називається допустимим розв'язком обмеження. Допустимий розв'язок обмеження (1.13), який є базисним, називається базисним допустимим розв'язком; якщо допустимий розв'язок є виродженим базисним розв'язком, то його називають виродженим базисним розв'язком.*

Тепер розглянемо задачу LP в канонічній формі

$$\begin{aligned} \min \quad & c \cdot x, \\ \text{за умов} \quad & Ax = b, \\ & x \geqslant 0. \end{aligned} \tag{1.14}$$

Допустимий розв'язок обмеження, який надає мінімального значення цільовій функції, називається *оптимальним допустимим розв'язком.* Якщо такий розв'язок базисний, то він називається *оптимальним базисним допустимим розв'язком.*

Теорема 1.7. *Нехай в задачі (1.14)* $\operatorname{rank} A = m$. *Тоді:*

(i) *якщо існує допустимий розв'язок, то існує базисний допустимий розв'язок;*

(ii) *якщо існує оптимальний допустимий розв'язок, то існує оптимальний базисний розв'язок.*

Теорема 1.8. *Нехай $A \in \mathbb{R}^{m\times n}$ і $\operatorname{rank} A = m$, $b \in \mathbb{R}^m$. Вектор x — екстремальна точка для $\mathcal{K} = \{x \in \mathbb{R}^n : Ax = b,\ x \geqslant 0\}$ — опуклого поліедра тоді і лише тоді, коли x — базисний допустимий розв'язок системи (1.13).*

За теоремою 1.7 розв'язок задачі LP потрібно шукати серед базисних допустимих розв'язків. Оскільки в задачі n змінних і m обмежень, то найбільше

$$\binom{n}{m} = \frac{n!}{m!(n-m)!}$$

базисних розв'язків. Отже, (за теоремою 1.8) для пошуку розв'язку задачі (1.14) достатньо перебрати лише екстремальні точки допустимої множини, кількість яких скінченна. Екстремальна точка в якій цільова функція $c \cdot x$ набуває найменшого значення, буде одним із розв'язків задачі (якщо задача має розв'язок). Одним із алгоритмів раціонального перебору, тобто перебору, за якого значення цільової функції зменшується при переході від однієї екстремальної точки до іншої, є симплекс-метод.

Існування і скінченна кількість екстремальних точок це частковий прояв загальних основ опуклого аналізу. Насправді, замкнена опукла множина має екстремальні точки тоді і лише тоді, коли вона не містить прямих. Позаяк множина вигляду (1.13) має таку властивість, оскільки є підмножиною в $\mathbb{R}^n_+$. Кожний поліедр $\mathcal{P} \neq \emptyset$ має скінченну кількість екстремальних точок.

Приклад 1.4. 1. Розглянемо задачу LP

$$\begin{aligned} \min\ & -x_1 - 2x_2, \\ \text{за умов}\ & -2x_1 + x_2 + x_3 = 2, \\ & -x_1 + x_2 + x_4 = 3, \\ & x_1 + x_5 = 3, \\ & x_i \geqslant 0,\ i = 1, \dots, 5. \end{aligned}$$

Наприклад, базисні змінні $\{x_2, x_3, x_5\}$, а базисний розв'язок $x = (0, 3, -1, 0, 3)$ відповідає недопустимій екстремальній точці; базис

$\{x_3, x_4, x_5\}$, а допустимий базисний розв'язок $x = (0, 0, 2, 3, 3)$; базис $\{x_1, x_2, x_3\}$, а оптимальний допустимий базисний розв'язок $x = (3, 6, 2, 0, 0)$.

2. Розглянемо обмеження

$$Ax = \begin{pmatrix} 2 & 1 & 0 & 0 \\ 3 & 0 & 1 & 0 \\ 4 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 6 \\ 13 \\ 12 \end{pmatrix} = b, \quad x \geqslant 0.$$

Базисна матриця

$$B = \begin{pmatrix} 2 & 1 & 0 \\ 3 & 0 & 1 \\ 4 & 0 & 0 \end{pmatrix},$$

а вироджений базисний допустимий розв'язок $(3, 0, 4, 0)$. ▲

1.3 LP і симплекс-метод

Оскільки задачу лінійного програмування і пошук оптимального розв'язку симплекс-методом вивчають у загальному курсі "Методи оптимізації і варіаційне числення", тому приділимо увагу лише тим темам, які не вивчаються або вивчають недостатньо. Насамперед розглянемо LP з двосторонніми обмеженнями. Для розгляду симплекс-методу її розв'язання, нагадаємо деякі означення.

Нехай задано LP:

$$\begin{aligned} &\max W(x) = \sum_{j=1}^{n} c_j x_j, \\ &\text{за умов} \qquad \sum_{j=1}^{n} a_{ij} x_j \leqslant b_i, \;\; i = 1, \ldots, m, \\ &\qquad\qquad x_j \geqslant 0, \;\; j = 1, \ldots, n. \end{aligned} \tag{1.15}$$

Задача (1.15) записана в канонічній формі, якщо обмеження $\sum_{j=1}^{n} a_{ij} x_j = b_i$, $i = 1, \ldots, m$, набули вигляду рівностей. Задачу

(1.15) завжди можна записати в канонічній формі, увівши слабкі змінні x_{n+i}, $i = 1, \ldots, m$, і перетворивши обмеження-нерівності в рівності. Звідси отримаємо задачу LP

$$\begin{aligned} &\max W(x) = \sum_{j=1}^{n} c_j x_j, \\ &\text{за умов} \qquad \sum_{j=1}^{n} a_{ij} x_j + x_{n+i} = b_i, \;\; i = 1, \ldots, m, \\ &\qquad\qquad x_j \geqslant 0, \;\; j = 1, \ldots, n+m. \end{aligned} \tag{1.16}$$

Будемо говорити, що задача (1.16) має *діагональну форму* стосовно змінних $x_{n+1}, \ldots, x_{n+m}$.

Розглянемо LP в канонічній формі

$$\begin{aligned} &\max W(x) = \sum_{j=1}^{n} c_j x_j, \\ &\text{за умов} \qquad \sum_{j=1}^{n} a_{ij} x_j = b_i, \;\; i = 1, \ldots, m, \\ &\qquad\qquad x_j \geqslant 0, \;\; j = 1, \ldots, n. \end{aligned} \tag{1.17}$$

Для розв'язку цієї задачі приймемо $I(x) = \{i : x_i \neq 0\}$. Якщо x — допустимий розв'язок задачі (1.17), то компоненти x_i, $i \in I(x)$, додатні, а решта компонент дорівнює нулю.

Означення 1.3. *Розв'язок $x^* = (x_1^*, \ldots, x_n^*)$, який задовольняє обмеження-нерівності задачі (1.17), називається базисним, якщо система стовпців $\{A_j : j \in I(x^*)\}$ матриці $A = (a_{ij})$ лінійно незалежна.*

Легко бачити, що ненульовий базисний розв'язок x^* — єдиний розв'язок системи рівнянь

$$\sum_{j \in I(x^*)} x_j A_j = b, \;\; x_i = 0, \; i \notin I(x^*).$$

Кількість базисних розв'язків скінчена. Надалі для базисних допустимих розв'язків будемо використовувати позначення БДР.

Нехай X — множина всіх допустимих розв'язків задачі (1.15). X — поліедр в $\mathbb{R}^n$. Обмежений поліедр — політоп.

Розв'язок $x^* \in X$ є БДР тоді і лише тоді, коли x^* — екстремальна (крайня) точка множини X.

Означення 1.4. *Базисом називається система* $\mathcal{B} = \{A_j : j \in I\}$, *яка складається з* m *лінійно незалежних стовпців матриці* A. *У цьому разі* I *будемо називати множиною базисних номерів* (*МБН*).

Очевидно, що $\mathcal{B}$ є базисом в $\mathbb{R}^m$ у звичайному алгебричному розумінні. Нехай x^* — базисний розв'язок. Кількість елементів $|I(x^*)|$ множини $I(x^*)$ не більше за m. Якщо $|I(x^*)| = m$, то розв'язок x^* невироджений, а в іншому випадку ($|I(x^*)| < m$) — вироджений. Невиродженому базисному розв'язку x^* відповідає єдиний базис $\mathcal{B} = \{A_j : j \in I(x^*)\}$. Для виродженого базисного розв'язку лінійно незалежну систему стовпців $\{A_j : j \in I(x^*)\}$ можна доповнити до базису $\mathcal{B} = \{A_j : j \in I\}$, $I(x^*) \subset I$. Для виродженого базисного розв'язку зазвичай відповідає декілька базисів.

Припустимо, що $b_i \geqslant 0$, $i = 1, \ldots, m$. Якщо задача LP у діагональній формі (1.15), то розв'язок $x^{(1)}$ вигляду

$$x^{(1)} = (\underbrace{0, \ldots, 0}_{n\,\text{раз}}, b_1, \ldots, b_m) \tag{1.18}$$

— базисний.

Симплекс-метод [4,5] застосовують для пошуку оптимального розв'язку LP. Він також дає змогу визначити сумісність обмежень задачі, а також чи є обмежена зверху цільова функція на множині допустимих розв'язків.

Оскільки серед розв'язків LP є екстремальні точки допустимої множини, кількість яких скінченна, то *головна ідея симплекс методу — раціональний перебір крайніх точок — у випадку задачі на максимум, значення цільової функції зростає при переході від однієї екстремальної точки до іншої.*

Зауважимо, що для задачі LP в діагональній формі (1.16) розв'язок $x^{(1)}$ вигляду (1.18) — базисний з МБН $I^{(1)} = \{n+1, \ldots, n+m\}$.

Означення 1.5. *Розв'язок x^*, який задовольняє обмеження-рівності задачі (1.16), називається реберним, якщо або $|I(x^*)| = r$ (випадок базисного розв'язку), або $|I(x^*)| = r+1$, де r — ранг системи стовпців $\{A_j : j \in I(x^*)\}$.*

Приклад 1.5. Для обмеження

$$x_1 + x_3 = 1,\ x_2 + x_4 = 1,\ \ x_i \geqslant 0,\ i = 1,2,3,4,$$

точки $(1, x_2, 0, x_4)$, де $x_2 + x_4 = 1$, $x_2, x_4 \geqslant 0$, — реберні розв'язки. ▲

Симплекс-метод застосований до двоїстої задачі **називають двоїстим**. Його особливість виявляється в тому, що перетворення виконують з симплекс-таблицею прямої задачі.

Розглянемо задачу LP в діагональній формі (1.16). Двоїсту задачу до неї можна записати у формі рівностей

$$\begin{aligned} &\min\ Z = \sum_{i=1}^{m} b_i y_i, \\ &\text{за умов} \quad \sum_{i=1}^{m} a_{ij} y_i - y_{m+j} = c_j,\ \ j = 1, \ldots, n, \\ &\qquad\qquad y_i \geqslant 0,\ \ i = 1, \ldots, m+n. \end{aligned} \tag{1.19}$$

Між змінними задач (1.16) і (1.19) є таке взаємно однозначне співвідношення

$$\begin{array}{c|c|c|c|c|c} x_1 & \ldots & x_n & x_{n+1} & \ldots & x_{n+m} \\ \hline y_{m+1} & \ldots & y_{m+n} & y_1 & \ldots & y_n \end{array}.$$

Початкова симплекс-таблиця з початковим БДР (1.18) і МБН

$I^{(1)} = \{n+1, \dots, n+m\}$ набула вигляду

W	$-c_1$	$\dots$	$-c_n$	0	$\dots$	0	0
x_{n+1}	a_{11}	$\dots$	a_{1n}	1	$\dots$	0	b_1
		$\vdots$		$\vdots$			
x_{n+m}	a_{m1}	$\dots$	a_{mn}	0	$\dots$	1	b_m

.

Для застосування двоїстого симплекс-методу необхідно, щоб її нульовий рядок містив невід'ємні елементи, тобто $-c_j \geqslant 0$, $j = 1, \dots, n$. Така симплекс-таблиця називається *двоїсто допустимою*. У цьому випадку

$$y^{(1)} = (\underbrace{0, \dots, 0}_{m \text{ раз}}, -c_1, \dots, -c_n)$$

— допустимий розв'язок задачі (1.19). Для розв'язку $y^{(1)}$ множина $J^{(1)} = \{m+1, \dots, n+m\}$ є МБН. Базисним змінним $y_{m+1}, \dots, y_{m+n}$ задачі (1.19) відповідають змінні $x_1, \dots, x_n$ задачі (1.16). Решта змінних $x_{n+1}, \dots, x_{n+m}$ є базисними змінними розв'язку $x^{(1)}$ вигляду (1.18) прямої задачі. Отож, МБН розв'язків $x^{(1)}$ і $y^{(1)}$ взаємно доповнюють один одного. Ці властивості зберігаються і на наступних кроках алгоритму. Алгоритм зупиняє роботу на k-му кроці, коли в поточній симплекс-таблиці виконується одна з таких умов:

1) $b'_i \geqslant 0$, $i = 1, \dots, m$; тоді поточні базисні розв'язки $x^{(k)}$ і $y^{(k)}$ оптимальні в задачах (1.16) і (1.19);

2) в опорному l-му рядку всі елементи a'_{ij} невід'ємні; у цьому випадку цільова функція задачі (1.19) не обмежена знизу.

Приклад 1.6. Розглянемо задачу LP

$$\begin{aligned} &\min W = 2x_1 + x_2, \\ &\text{за умов} \qquad x_1 - 2x_2 \leqslant 2,\ 3x_1 - x_2 \geqslant 1,\ x_1, x_2 \geqslant 0. \end{aligned}$$

Зведемо задачу до вигляду (1.16)

$$\begin{aligned} &\max W = -2x_1 - x_2, \\ &\text{за умов} \qquad x_1 - 2x_2 + x_3 = 2,\ -3x_1 + x_2 + x_4 = 1,\ x_1, x_2, x_3, x_4 \geqslant 0. \end{aligned}$$

Початкова симплекс-таблиця така:

W	2	1	0	0	0
x_3	1	-2	1	0	2
x_4	$\underline{-3}$	1	0	1	-1

.

У цій таблиці нульовий рядок невід'ємний, тому $y^{(1)} = (0, 0, 2, 1)$ — початковий базисний розв'язок. Опорний другий рядок і перший стовпець визначається однозначно. Після перетворення таблиці отримаємо

	y_3	y_4	y_1	y_2	
W	0	5/3	0	2/3	$-2/3$
x_3	0	$-5/3$	1	1/3	5/3
x_4	1	$-1/3$	0	$-1/3$	1/3

.

Оскільки $b'_i > 0$, $i = 1, 2$, то $y^{(2)} = (0, 2/3, 0, 5/3)$ — оптимальний розв'язок задачі (1.19), $x^{(2)} = (1/3, 0, 5/3, 0)$ — оптимальний розв'язок задачі (1.16), а $x^* = (1/3, 0)$ — оптимальний розв'язок вихідної задачі LP. Оскільки в оптимальному розв'язку двоїстої задачі $y_4^* > 0$, $y^*2 > 0$, то за властивістю доповнювальної нежорсткості в кожному оптимальному розв'язку прямої задачі (1.16), відповідно, друга і четверта компоненти дорівнюють нулю. Тому знайдений розв'язок єдиний. ▲

Означення 1.6. *Ненульовий вектор $a \in \mathbb{R}^s$ називається лексикографічно додатним (позначення: $a \succ 0$), якщо перша його ненульова координата більша за нуль. Вектор $a \in \mathbb{R}^s$ називається лексикографічно від'ємним (позначення: $a \prec 0$), якщо $-a \succ 0$. Вектор $a \in \mathbb{R}^s$ лексикографічно більший за вектор $b \in \mathbb{R}^s$ (позначення: $a \succ b$), якщо $a - b \succ 0$.*

Нехай a_{0j} — j-та компонента нульового рядка симплекс-таблиці, а її стовпці позначимо через

$$\overline{A}_j = (a_{ij},\, i = 0, \ldots, m)^T,\ j = 1, \ldots, n + m.$$

Означення 1.7. *Симплекс-таблиця називається строго двоїсто допустимою, якщо всі стовпці* $\overline{A}_j$, $j = 1, \ldots, n+m$, *лексиграфічно додатні.*

Позначимо через $lex \max\limits_{1 \leqslant j \leqslant n} a^{(j)}$ лексикографічний максимум векторів $a^{(j)}$, $j = 1, \ldots, n$.

Теорема 1.9. *Нехай симплекс-таблиця строго двоїсто допустима. Припустимо, що в двоїстому симплекс-алгоритмі опорний* l*-й рядок вибирають довільно з умови* $b_l < 0$*, а опорний* p*-й стовпець — із рівності*

$$lex \max_{j:\, a_{ij}<0} \frac{\overline{A}_j}{a_{ij}} = \frac{\overline{A}_p}{a_{lp}}.$$

Якщо тут лексикографічних стовпців декілька, то серед них будемо вибирати стовпець з найбільшим номером. Тоді двоїстий симплекс-алгоритм збігається за скінченну кількість кроків, а в процесі ітерацій останній стовпець симплекс-таблиці лексикографічно спадає.

Задачі

1.1 Задачу LP

$$\begin{aligned} & \min W = 7x_1 + 6x_2 + 5x_3, \\ \text{за умов} \quad & 4x_1 + x_2 + 2x_3 \geqslant 10, \\ & 3x_1 + 8x_2 + 6x_3 \geqslant 18, \\ & x_1 + 2x_2 + 6x_3 \geqslant 15,\ x_1, x_2, x_3 \geqslant 0, \end{aligned}$$

сформулюйте як задачу на максимум і застосуйте двоїстий симплекс-метод. Опорний рядок вибирайте з найменшим можливим номером $l = \min\{i : b'_i < 0\}$, а опорний стовпець так, як це показано в формулюванні теореми 1.9.

Відповідь. Оптимальний розв'язок $x^* = (4/3, 0, 7/3)$.

1.2 Двоїстим симплекс-методом розв'язати такі задачі LP:

1)

$$\begin{aligned} \min W &= 2x_1 + 3x_2, \\ \text{за умов} \quad & 4x_1 + x_2 \geqslant 5, \\ & 3x_1 + 4x_2 \geqslant 1, \\ & x_1 + 3x_2 \geqslant 4, \\ & x_j \geqslant 0,\ j = 1, 2; \end{aligned}$$

2)

$$\begin{aligned} \min W &= 3x_1 + 2x_2 + 4x_3 + 3x_4, \\ \text{за умов} \quad & 2x_1 + 3x_2 + 4x_3 + 2x_4 \geqslant 1, \\ & 4x_1 + 2x_2 + 3x_3 + 4x_4 \geqslant 1, \\ & x_j \geqslant 0,\ j = 1, 2, 3, 4. \end{aligned}$$

Відповідь. 1) $x^* = (1, 1)$; 2) $x^* = (1/8, 1/4, 0, 0)$.

1.3 На заводі треба вирішити, яку кількість x_1 чистої сталі і яку кількість x_2 брухту потрібно використати для виготовлення (з відповідного сплаву) виливка для одного з замовників. Нехай виробничі витрати на 1 т сталі становить 3 у. о., а витрати на 1 т брухту — 5 у. о. (останнє число більше попереднього, позаяк використання брухту пов'язане з його попереднім очищенням). Замовлення передбачає постачання не менше 5 т виливка; у цьому разі замовник готовий купити і велику кількість виливка, якщо завод поставить перед ним таку умову. Відношення маси брухту до маси чистої сталі в сплаві не повинно перевищувати $7/8$. Технологічні умови такі, що на процеси плавлення і виливання не може бути відведено більше ніж 18 год, зокрема на 1 т сталі витрачається 3 год, а на 1 т брухту — 2 год. Мета заводу — виконати замовлення з мінімальними виробничими витратами. Складіть задачу LP на максимум і розв'яжіть її двоїстим симплекс-методом.

Відповідь. $x^* = (5, 0)$.

1.4 Аналіз чутливості

Крім наведених в п. 1.1 співвідношень двоїстості, є також і інші, одним з яких є така теорема.

Теорема 1.10. *Нехай $x^* = x^*(b)$ — оптимальний невироджений базисний розв'язок задачі (1.17), який відповідає вектору b. Тоді оптимальний розв'язок двоїстої задачі єдиний і його координати визначаються рівностями*

$$\lambda_i^* = \frac{\partial(c \cdot x^*(b))}{\partial b_i}, \quad i = 1, \ldots, m. \tag{1.20}$$

У термінах виробничої задачі (див. приклад 1.2) рівності (1.20) означають, що λ_i^* — степінь (міра) чутливості максимального прибутку до зміни i-го чинника: якщо $\lambda_i^* > 0$, то збільшення обсягу i-го чинника призводить до збільшення максимального прибутку і тим ефективніше, чим більше λ_i^*; при $\lambda^* < 0$ до збільшення максимального прибутку призводить зменшення обсягу i-го чинника. Із (1.20) отримаємо, що приріст максимального значення цільової функції дорівнює

$$\triangle W_{\max} = \lambda_1^* \triangle b_1 + \cdots + \lambda_m^* \triangle b_m. \tag{1.21}$$

Зауважимо, що все наведене є правильним не при будь-яких змінах чинників, а лише за деяких в околі заданого значення вектора b. Отож, для проведення аналізу розв'язку LP треба знати оптимальний двоїстий розв'язок.

Для базисного розв'язку (див. п.1.2) головне обмеження $Ax = b$ набуває вигляду $Bx_B = b$. Тому базисний розв'язок $\overline{x}$ можна побудувати за базисною матрицею B: $x_B = B^{-1}b \geqslant 0$, $\overline{x} = (x_B, 0)$. Використаємо це для аналізу впливу змін чинників на зміну оптимального значення цільової функції, тобто для аналізу чутливості до вектора чинників. Всі зміни вектора ресурсів b повинні бути такими, щоб $B^{-1}(b + \triangle b) \geqslant 0$, де B — базисна матриця, яка відповідає оптимальному розв'язку x^*. У разі виконання умови $B^{-1}(b + \triangle b) \geqslant 0$ оптимальний двоїстий розв'язок λ^* буде залишатися тим самим (**стійким до зміни вектора**

ресурсів). Оптимальний розв'язок $x^*(b + \triangle b)$, який відповідає вектору ресурсів $b + \triangle b$, а також максимальне значення $W_{\max}$ у цьому разі може змінитися, але базис і двоїстий оптимальний розв'язок λ^* залишаються незмінними.

Приклад 1.7. Виробнича задача. Підприємство виготовляє два види продукції q_1, q_2. Для їхнього виготовлення необхідно витратити такі чинники: сировину, фізичну працю та знання. Витрати чинників на одиницю продукції кожного виду, відповідно: 8,8,1 і 25,5,5. Обсяг наявних ресурсів на день, відповідно 800,640 і 145. Прибуток на одиницю продукції кожного виду 80 і 70. Треба скласти план випуску продукції за день, за якого прибуток буде максимальним.

Побудуємо математичну модель. Позначимо через x_j кількість одиниць продукції вигляду q_j, $j = 1, 2$, які підприємство виготовить за добу. Тоді $80x_1$ — прибуток від реалізації продукції вигляду q_1, $70x_2$ — вигляду q_2, $W(x) = 80x_1 + 70x_2$ — прибуток від реалізації всієї продукції. Запишемо в моделі обмеження на чинники. Маємо: $8x_1$ — обсяг сировини, яка необхідна для виготовлення продукції вигляду q_1, $25x_2$ — вигляду q_2, $8x_1+25x_2$ — всієї продукції. Оскільки в наявності є 800кг сировини, то маємо $8x_1+25x_2 \leqslant 800$. Аналогічно, обмеження на працю: $8x_1+5x_2 \leqslant 640$, обмеження на знання $x_1+5x_2 \leqslant 145$. Із фізичного змісту змінних випливає, що всі вони невід'ємні $x_j \geqslant 0$.

Виробнича задача звелася до розв'язання такої математичної задачі:

$$\begin{aligned} \max W(x) = & 80x_1 + 70x_3, \\ \text{за умов} \quad & 8x_1 + 25x_2 \leqslant 800, \\ & 8x_1 + 5x_2 \leqslant 640, \\ & x_1 + 5x_2 \leqslant 145, \\ & x_i \geqslant 0,\ i = 1, 2. \end{aligned} \tag{1.22}$$

Задача (1.22) — **математична модель виробничої задачі.**

Зведемо її до канонічного вигляду

$$\begin{aligned} \max W(x) =& 80x_1 + 70x_3, \\ \text{за умов} \quad & 8x_1 + 25x_2 + x_3 = 800, \\ & 8x_1 + 5x_2 + x_4 = 640, \\ & x_1 + 5x_2 + x_5 = 145, \\ & x_i \geqslant 0,\ i = 1, \ldots, 5. \end{aligned} \tag{1.23}$$

Для цієї задачі $x = (x_1, x_2, x_3, x_4, x_5)$, $c = (80, 70, 0, 0, 0)$, $b = (800, 640, 145)$ і

$$A = \begin{pmatrix} 8 & 25 & 1 & 0 & 0 \\ 8 & 5 & 0 & 1 & 0 \\ 1 & 5 & 0 & 0 & 1 \end{pmatrix}.$$

Змінні x_3, x_4, x_5 — слабкі. Вихідним базисним (невиродженим) розв'язком є вектор $\overline{x} = (0, 0, 800, 640, 145)$ з базисною матрицею $B = (A_3, A_4, A_5)$ і $I = \{3, 4, 5\}$. Симплекс-методом знаходимо оптимальний базисний розв'язок $x^* = (75, 8, 0, 0, 30)$. Для вихідної задачі $x_1^* = 75$, $x_2^* = 8$. Максимальне значення $W_{\max} = 6560$. Величина $x_5^* = 30$ для виробничої задачі означає невикористаний чинник, а в нашій задачі — це 30 годин невикористаної інтелектуальної праці при випуску обох видів продукції.

Запишемо тепер двоїсту задачу

$$\begin{aligned} \min Z(\lambda) =& 800\lambda_1 + 640\lambda_2 + 145\lambda_3, \\ \text{за умов} \quad & 8\lambda_1 + 8\lambda_2 + \lambda_3 \geqslant 80, \\ & 25\lambda_1 + 5\lambda_2 + 5\lambda_3 \geqslant 70, \\ & x_1 + 5x_2 \leqslant 145, \\ & \lambda_i \geqslant 0,\ i = 1, 2, 3. \end{aligned} \tag{1.24}$$

Оскільки відомий розв'язок прямої задачі, то розв'язок двоїстої задачі можна знайти за допомогою умов доповняльності (1.10): $\lambda^* = (1, 9, 0)$.

Оптимальному базисному розв'язку відповідає матриця $B =$

(A_2, A_1, A_5), а відповідна

$$B^{-1} = \begin{pmatrix} 1/20 & -1/20 & 0 \\ -1/32 & 5/32 & 0 \\ -7/32 & 3/32 & 1 \end{pmatrix}.$$

Базисні координати базисного розв'язку — це координати вектора b в поточному базисі, тобто $x_B = B^{-1}b = (75, 8, 30)$. Границі зміни вектора чинників, для яких двоїстий оптимальний план залишається незмінним, знаходять як розв'язок нерівності

$$B^{-1}(b + \triangle b) = \begin{pmatrix} 1/20 & -1/20 & 0 \\ -1/32 & 5/32 & 0 \\ -7/32 & 3/32 & 1 \end{pmatrix} \begin{pmatrix} 800 + \triangle b_1 \\ 640 + \triangle b_2 \\ 145 + \triangle b_3 \end{pmatrix} \geqslant 0.$$

1. Прийнявши $\triangle b_2 = \triangle b_3 = 0$, з першої нерівності отримаємо $\triangle b_1 \geqslant -160$, з другої — $\triangle b_1 \leqslant 2400$, а з третьої — $\triangle b_1 \leqslant 137\frac{1}{7}$, тобто при незмінних чинниках b_2 і b_3 та зміни b_1 в межах від 640 до $937\frac{1}{7}$ двоїстий оптимальний розв'язок залишається незмінним, оскільки $\lambda_1^* = 1 > 0$, то зі збільшенням сировини від 800 до $937\frac{1}{7}$ кг максимальний прибуток буде збільшуватися, а зі зменшенням від 800 до 640кг — зменшується.

2. При $\triangle b_1 = 0$, $\triangle b_3 = 0$ знаходимо: $-320 \leqslant \triangle b_2 \leqslant 160$, тобто при незмінних b_1 і b_3 і $320 \leqslant b_2 \leqslant 800$ оптимальний двоїстий розв'язок не змінюється, оскільки $\lambda_2^* = 9 > 0$, то як у першому випадку збільшення праці від 640 до 800 г призведе до збільшення максимального прибутку, а зменшення від 640 до 320 г — до зменшення максимального прибутку.

3. При $\triangle b_1 = 0$, $\triangle b_2 = 0$ знаходимо:$\triangle b_2 \geqslant -30$. Отож, за незмінних b_1 і b_2 і $115 \leqslant b_3 < \infty$ оптимальний двоїстий розв'язок є незмінним, оскільки $\lambda_3^* = 0$, то в цьому випадку максимальний прибуток не змінюється.

Можна дослідити і одночасну зміну всіх трьох координат вектора b, за якого оптимальний двоїстий розв'язок залишається незмінним. ▲

Можна також проводити *аналіз чутливості у випадку зміни вектора вартості.*

Уведемо функцію $\gamma(c) = \max\{c \cdot x : x \in X\}$, де X — множина розв'язків задачі LP. Позначимо через $X^*(c)$ — множину оптимальних розв'язків. Функція $\gamma(c)$ характеризує залежність максимального значення цільової функції від вектора вартості. Множина $X^*(c)$ при кожному c — непорожній опуклий компакт.

Позначимо через

$$\nu_i^+ = \frac{\partial\gamma(c)}{\partial c_i^+}, \quad \nu_i^- = \frac{\partial\gamma(c)}{\partial c_i^-} \tag{1.25}$$

праву та ліву частинні похідні за c_i від функції $\gamma(c)$. Числа ν_i^+ і ν_i^- називають **правим і лівим коефіцієнтами чутливості** цільової функції за i-ю компонентою вектора вартості.

Фізичний зміст коефіцієнтів чутливості випливає із означення (1.25): $\nu_i^+(\nu_i^-)$ — швидкість зміни максимального значення цільової функції у випадку зміни компоненти c_i (решта компонент вектора c фіксовані) в правому (лівому) околі вихідного значення.

Теорема 1.11. *Коефіцієнт чутливості $\nu_i^+(\nu_i^-)$ дорівнює максимальній (мінімальній) i-й координаті серед всіх оптимальних розв'язків LP*

$$\nu_i^+ = \max_{x_i \in X_i^*(c)} x_i, \quad \nu_i^- = \min_{x_i \in X_i^*(c)} x_i, \ \ i = 1, \ldots, n,$$

де $X_i^*(c)$ — проекція множини $X^*(c)$ на вісь Ox_i. Якщо оптимальний розв'язок x^* єдиний, то $\nu_i^+ = \nu_i^- = x_i$, $i = 1, \ldots, n$.

У прикладі 1.7 оптимальний план $x^* = (75, 8)$ єдиний, тому $\nu_1^+ = \nu_1^- = \nu_1 = 75$, $\nu_2^+ = \nu_2^- = \nu_2 = 8$. Зі змісту коефіцієнтів чутливості випливає, що максимальний прибуток підприємства в 9,375 раза $(\nu_1 : \nu_2)$ чутливіший до зміни прибутку на одиницю продукції першого вигляду, ніж другого.

1.5 LP з двосторонніми обмеженнями

Розглянемо задачу

$$\begin{aligned} &\max W(x) = \sum_{j=1}^{n} c_j x_j, \\ &\text{за умов} \quad \sum_{j=1}^{n} a_{ij} x_j \leqslant b_i, \;\; i = 1, \ldots, m, \\ &\qquad 0 \leqslant x_j \leqslant u_j, \;\; j = 1, \ldots, n. \end{aligned} \tag{1.26}$$

Порівняно з задачею (1.15), на змінні накладено деякі додаткові обмеження $x_j \leqslant u_j$, де u_j, $j = 1, \ldots, n$ – задані додатні числа. Подібні задачі з двосторонніми обмеженнями виникають, наприклад, у тих випадках, коли вихідною множиною розв'язків є паралелепіпед. Такі задачі трапляються під час розв'язання задач цілочислового LP методом віток і границь.

Приклад 1.8. Оптимізація структури страхового портфеля. Медична страхова компанія проводить страхування за двома видами захворювань, які потребують хірургічних операцій. Ймовірність першого захворювання — $0,0048$, а термін перебування в лікарні становить в середньому 30 днів. Аналогічні цифри при другому захворюванні, відповідно $0,0018$ і 60. Компанія може забезпечити в рік 240 і 144 операції, відповідно, при першому і другому захворюванні, а страхові премії (внески) становлять 2 і 3 у.о. Компанія має 32 лікарняних ліжка. Потрібно визначити оптимальний вміст страхового портфеля, який забезпечує максимальну страхову премію при заданих обмеженнях.

Сформулюємо задачу LP. Нехай x_j — число десятків тисяч договорів, які укладаються за j-м видом страхування, $j = 1, 2$. Для першого виду страхування очікувана кількість страхових випадків $10^4 x_1 \cdot 0,0048 = 48x_1$ не повина перевищувати 240. Тому отримуємо нерівність $x_1 \leqslant 5$. Аналогічні міркування для другого виду страхування приводять до нерівності $x_2 \leqslant 8$. Оскільки протягом року в одному номері лікарні можна розмістити 12 па-

цієнтів з першим захворюванням або 6 пацієнтів з другим захворюванням, то отримаємо ще одне обмеження

$$\frac{48x_1}{12} + \frac{18x_2}{6} \leqslant 32.$$

Отже, одержуємо таку задачу LP:

$$\begin{aligned} &\min W = 2x_1 + 3x_2, \\ &\text{за умов} \quad 4x_1 + 3x_2 \leqslant 32,\ \ 0 \leqslant x_1 \leqslant 5,\ \ 0 \leqslant x_2 \leqslant 8. \end{aligned}$$

▲

Сформулюємо задачу з двосторонніми обмеженнями в діагональній формі

$$\begin{aligned} &\max W(x) = \sum_{j=1}^{n} c_j x_j, \\ &\text{за умов} \quad \sum_{j=1}^{n} a_{ij} x_j + x_{n+i} = b_i,\ \ i = 1, \ldots, m, \\ &\qquad 0 \leqslant x_j \leqslant u_j,\ \ j = 1, \ldots, n,\ \ x_{n+i} \geqslant 0,\ \ i = 1, \ldots, m. \end{aligned} \tag{1.27}$$

Зауважимо, що на слабкі змінні x_{n+i} не накладаються двосторонні обмеження. Можна ввести додаткові слабкі змінні та перетворити нерівності $x_j \leqslant u_j$ в рівності, але у цьому разі збільшиться розмірність симплекс-таблиці. Позаяк цього можна не робити, а використати звичайну симплекс-таблицю задачі (1.16).

Надалі припустимо, що $b_i \geqslant 0$, $i = 1, \ldots, m$. Як і у випадку розв'язування задачі (1.16), це дає змогу почати роботу симплекс-алгоритму з БДР $x^{(1)}$ вигляду (1.18).

Поряд із змінними x_j, $i = 1, \ldots, n$, будемо розглядати доповнювальні змінні $y_j = u_j - x_j$, $i = 1, \ldots, n$. Легко бачити, що $0 \leqslant y_j \leqslant u_j$. Якщо в задачі (1.16) деякі змінні x_j замінити на $u_j - y_j$, то отримуємо еквівалентну задачу. Такі перетворення будемо використовувати в процесі застосування симплекс-методу.

Розглянемо як перетвориться симплекс-таблиця у випадку заміни змінної $y_j = u_j - x_j$.

Розглянемо симплекс-таблицю на довільному кроці алгоритму

W	$-c'_1$	$\dots$	$-c'_{n+m}$	c'_0
$x_{B(1)}$	a'_{11}	$\dots$	a'_{1n+m}	b'_1
		$\vdots$		$\vdots$
$x_{B(m)}$	a'_{m1}	$\dots$	a'_{mn+m}	b'_m

.

Заміна змінної x_j на y_j призводить до того, що з останнього стовпця симплекс-таблиці віднімається j-й, помножений на u_j, а елементи j-го стовпця змінюють знак.

Позначимо через $x^{(k)}$ поточний базисний розв'язок з базисними компонентами $x^{(k)}_{B(i)} = b'_i$, $i = 1, \dots, m$. Нехай $-c'_p < 0$, тобто не виконується умова оптимальності. Будемо вводити в базис змінну x_p (або y_p, якщо до цього була зроблена заміна $y_p = u_p - x_p$). Як завжди, перехід до базисного розв'язку $x^{(k+1)}$ виконується збільшенням компоненти x_p від нуля. Зокрема, реберний розв'язок набув вигляду

$$\tilde{x} = (\dots, 0, \dots, x_p, \dots, \underbrace{b'_i - x_p}_{B(i)} a'_{ip}, \dots, 0, \dots).$$

Збільшуємо x_p доти, доки розв'язок $\tilde{x}$ не перестане бути допустимим в задачі (1.26). Щоб компоненти розв'язку $\tilde{x}$ були невід'ємними, необхідно виконати нерівності

$$x_p \leqslant \theta = \min_{i:\, a'_{ip} > 0} \frac{b'_i}{a'_{ip}} = \frac{b'_l}{a'_{lp}}.$$

Також треба стежити за виконанням нерівностей вигляду

$$b'_i - x_p a'_{ip} \leqslant u_{B(i)}, \text{ де } B(i) \leqslant n.$$

Остання нерівність може не справджуватися лише при $a'_{ip} < 0$. Звідси одержимо

$$x_p \leqslant \theta_1 := \min_{i:\, a'_{ip} < 0,\, B(i) \leqslant n} \frac{u_{B(i)} - b'_i}{-a'_{ip}} = \frac{u_{B(l)} - b'_l}{-a'_{lp}}.$$

Якщо $a'_{ip} \geqslant 0$ для всіх i, таких, що $B(i) \leqslant n$, тоді умовно можна прийняти $\theta_1 = +\infty$. Отже, x_p не може перевищувати кожну з трьох величин: u_p, θ, θ_1, тобто $x_p \leqslant \min[u_p, \theta, \theta_1]$.

Розглянемо три можливі випадки.

1. $\theta \leqslant [u_p, \theta_1]$. Тоді x_p збільшуємо до величини $\theta = b'_l/a'_{lp}$, де l – номер опорного рядка. Виконаємо операцію заміщення, увівши в базис змінну x_p (або y_p) і вивівши з нього змінну $x_{B(l)}$.

2. $\theta_1 \leqslant [u_p, \theta]$. У цьому випадку компоненту x_p збільшуємо до величини

$$\theta_1 = \frac{u_{B(l)} - b'_l}{-a'_{lp}}.$$

У цьому разі $B(l)$-я компонента розв'язку $\tilde{x}$ набуде значення $u_{B(l)}$. Виконаємо заміну змінної $y_{B(l)} = u_{B(l)} - x_{B(l)}$. Оскільки в розв'язку $\tilde{x}$, $x_{B(l)} = u_{B(l)}$, то $y_{B(l)} = 0$. Отож, змінна $y_{B(l)}$ буде виведена з базису. Тому після заміни змінної треба виконати операцію заміщення з опорним елементом a'_{lp}. Легко перевірити, що при цих перетвореннях у розв'язку $x^{(k+1)}$ базисна координата x_p дорівнюватиме θ_1. Справді, для заміни змінної $y_{B(l)} = u_{B(l)} - x_{B(l)}$ з останнього стовпчика симплекс-таблиці віднімається базисний стовпчик e_l, помножений на $u_{B(l)}$. Тому в останньому стовпчику зміниться тільки один l-й елемент, який буде дорівнювати $b'_l - u_{B(l)}$. Після операції заміщення він буде дорівнювати $(b'_l - u_{B(l)})/a_{lp} = \theta_1$.

3. $u_p \leqslant \min[\theta, \theta_1]$. В розв'язку $\tilde{x}$ приймемо $x_p = u_p$ і виконаємо заміну змінної $y_p = u_p - x_p$. Тоді отримаємо, що $y_p = 0$. У цьому випадку базис залишається незмінним, але таблицю потрібно перетворити відповідно до заміни змінної.

Після завершення перетворень перевіряється умова оптимальності, тобто *невід'ємність нульового рядка симплекс-таблиці*. Якщо вона виконується, то розв'язок $x^{(k+1)}$ оптимальний. Інакше, ітерацію алгоритму продовжуємо.

Зауважимо таке: якщо в вихідній задачі (1.26) двосторонні обмеження накладаються лише на частину змінних x_j, то наведені міркування відповідно модифікуються.

Приклад 1.9. Розглянемо задачу (LP)

$$\begin{aligned} &\max W = 2x_1 + x_2 + 2x_3, \\ \text{за умов} \qquad &4x_1 + x_2 = 12; \quad -2x_1 + x_3 = 4, \\ &0 \leqslant x_1 \leqslant 4, \quad 0 \leqslant x_2 \leqslant 15, \quad 0 \leqslant x_3 \leqslant 6. \end{aligned}$$

Початкова симплекс-таблиця набуде вигляду

W	-2	-1	-2	0
x_2	4	1	0	12
x_3	-2	0	1	4

.

Забезпечимо виконання умови симплекс-алгоритму — цільова функція W залежить лише від небазисних:

W	-2	0	0	20
x_2	$\underline{4}$	1	0	12
x_3	-2	0	1	4

,

де

$$p = 1, \quad u_1 = 4, \quad \theta = \frac{12}{4} = 3, \quad \theta_1 = \frac{u_{B(2)} - b_2}{-a_{21}} = \frac{6-4}{2} = 1.$$

Оскільки $\theta_1 < \min[\theta, u_1]$, то потрібно виконати заміну змінної $x_3 = 6 - y_3$ і провести операцію заміщення з опорним елементом a'_{21}:

W	-2	0	0	20
x_2	4	1	0	12
y_3	$\underline{-2}$	0	-1	-2

, | W | 0 | 0 | 1 | 22 |
|---|---|---|---|---|
| x_2 | 0 | 1 | -2 | 8 |
| x_1 | 1 | 0 | $1/2$ | 1 |

.

Звідси знаходимо, що $x^* = (x_1^*, x_2^*, y_3^*) = (1, 8, 0)$ оптимальний розв'язок і $x_3^* = 6$. ▲

Задачі

1.4 Розв'язати задачі (LP) з двосторонніми обмеженнями:

1)

$$\begin{aligned} \max W &= 9x_1 + 6x_2 + 4x_3, \\ \text{за умов} \quad & 9x_1 + 3x_2 + 2x_3 \leqslant 12, \\ & 3x_1 + 6x_2 + x_3 \leqslant 10, \\ & 0 \leqslant x_j \leqslant 1,\ j = 1, 2, 3; \end{aligned}$$

2)

$$\begin{aligned} \max W &= x_1 + x_2, \\ \text{за умов} \quad & 2x_1 + x_2 \leqslant 2, \\ & x_1 + 3x_2 \leqslant 3, \\ & 3x_1 + 2x_2 \leqslant 5 \\ & 0 \leqslant x_j \leqslant 1,\ j = 1, 2, ; \end{aligned}$$

3)

$$\begin{aligned} \max W &= x_1 + 2x_2 + 2x_3 + 2x_4, \\ \text{за умов} \quad & 2x_1 + x_2 + 4x_3 + 2x_4 \leqslant 3, \\ & 4x_1 + 3x_2 + x_3 + 2x_4 \leqslant 3, \\ & 0 \leqslant x_j \leqslant 1,\ j = 1, 2, 3, 4. \end{aligned}$$

Відповідь. 1) $x^* = (7/9, 1, 1)$; 2) $x^* = (3/5, 4/5)$;, 3) $x^* = (0, 3/11, 2/11, 1)$.

1.5 В умові задачі 2.3 додатково припустимо, що запаси чистої сталі, які призначені для лиття, становлять 4 т, а брухту — 6 т. Розв'язати задачу LP з двосторонніми обмеженнями.

Відповідь. $x^* = (4, 1)$.

1.6 Розв'язати задачу з прикладу 1.8 двома способами: звичайним симплекс-методом і як задачу лінійного програмування з двосторонніми обмеженнями, а також надати інтерпретацію розв'язку.

Відповідь. $x^* = (2, 8)$.

1.6 Транспортна задача

Багато задач LP формулюють як транспортні задачі або зводять до них. Сформулюємо транспортну задачу. Нехай є m пунктів виробництва деякого продукту. Нехай в i-му пункті його випускають у кількості $s_i,\ i = 1, \ldots, m$. Продукт може бути неподільним (у цьому випадку величини s_i — цілі) та нескінченно подільним (наприклад, водні ресурси). Нехай є n пунктів споживання продукту з потребами $d_j,\ j = 1, \ldots, n$. Припустимо, що виконується *умова балансу*

$$\sum_{i=1}^{m} s_i = \sum_{j=1}^{n} d_j.$$

Нехай деяка транспортна компанія проводить перевезення продукту з пункту виробництва в пункти споживання. Позначимо через c_{ij} вартість перевезення одиниці продукту з i-го пункту виробництва в j-й пункт споживання. Зазвичай вартість перевезення пропорційна відстані між пунктами.

План перевезення можна зобразити матрицею $X = (x_{ij})_{ij}^{mn}$. Згідно з цим планом з пункту i продукт перевозиться в пункт j в кількості x_{ij} одиниць. Отож, маємо таку задачу мінімізації вартості перевезення:

$$\begin{aligned} & \min W = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij}, \\ \text{за умов} \quad & \sum_{j=1}^{n} x_{ij} = s_i, \quad i = 1, \ldots, m, \\ & \sum_{i=1}^{m} x_{ij} = d_j, \quad j = 1, \ldots, n, \\ & x_{ij} \geqslant 0, \quad i = 1, \ldots, m, \quad j = 1, \ldots, n. \end{aligned} \tag{1.28}$$

Задачу (1.28) називають *стандартною транспортною задачею.*

Часто виникають транспортні задачі, в яких умова балансу порушена. За умов дефіциту продукту виконується нерівність

$$\sum_{i=1}^{m} s_i < \sum_{j=1}^{n} d_j.$$

У цьому випадку також можна мінімізувати вартість перевезення, але деякі пункти споживання отримають продукт не в повних обсягах. Зведемо цю задачу до стандартної. Введемо $(m+1)$-й фіктивний пункт виробництва з величиною випуску

$$s_{m+1} = \sum_{j=1}^{n} d_j - \sum_{i=1}^{m} s_i.$$

У цьому разі варто прийняти $c_{m+1,j} = 0$, $j = 1, \ldots, n$ і розглянути (1.28) з $m+1$ пунктами виробництва. Нехай x^* — оптимальний розв'язок цієї задачі. Якщо $x^*_{m+1,j} > 0$, тоді в j-й пункт споживання буде недопоставлена продукція в кількості $x^*_{m+1,j}$. Якщо треба, щоб в j_1-й пункт була повна доставка, тоді доцільно прийняти $c_{m+1,j_1} = M$, де M — достатньо велике додатне число. Можна розглянути ще один варіант формулювання задачі, *коли компанія платить штраф величиною c_j одиниць за кожну недопоставлену в j-й пункт одиницю продукції*. Тоді потрібно прийняти $c_{m+1,j} = c_j$. Може виникнути задача, коли перевезення з i-го пункту в j-й неможливе (наприклад, при постачанні води з водойм до міста). Тут також варто застосувати "метод великого M" і прийняти $c_{ij} = M$.

Припустимо, що є надлишок продукту, тобто

$$\sum_{i=1}^{m} s_i > \sum_{j=1}^{n} d_j.$$

Для зведення задачі до стандартної форми (1.28) уведемо фіктивний $(n+1)$-й пункт споживання і покладемо

$$d_{m+1} = \sum_{i=1}^{m} s_i - \sum_{j=1}^{n} d_j, \quad c_{i,n+1} = 0,\ i = 1, \ldots, m.$$

Якщо x^* — оптимальний розв'язок цієї задачі і $x^*_{i,n+1} > 0$, тоді в i-му пункті виробництва не візьмуть продукту кількістю $x^*_{i,n+1}$. Як і в випадку дефіциту продукту, тут можливі інші методи визначення величин $c_{i,n+1}, i = 1, \ldots, m$.

Наведемо два приклади задач LP, які в початковому формулюванні не є транспортними задачами, але за змістом є такими.

Задача про призначення

Завод має n цехів і закупив m станків, $m \leqslant n$. У кожному цеху можна встановити не більше одного станка. Нехай c_{ij} — вартість встановлення i-го станка в j-му цеху. Потрібно розмістити станки в цехах з найменшими витратами на їхнє встановлення. Цю задачу можна розглянути як транспортну з $d_j = s_i = 1$ при всіх i, j. Якщо $m < n$, тоді маємо задачу з дефіцитом продукту.

Транспортна задача, яку побачимо далі, має таку властивість. Якщо всі величини s_j і d_j — цілі числа, то існує оптимальний розв'язок з цілими координатами. Тому координати такого оптимального розв'язку задачі про призначення дорівнюють нулю або одиниці.

Задача про графік доставок

Фірма повинна виконати доставку виробів протягом n місяців у кількості $d_1, \ldots, d_n$ одиниць. Фірма може в i-му місяці виготовити s_i виробів. Припустимо, що

$$\sum_{i=1}^{j} s_i > \sum_{i=1}^{j} d_i, \;\; j = 1, \ldots, n.$$

У деякі місяці можна виготовити виробів більше, ніж потрібно, але тоді частину продукції доведеться зберігати на складах з оплатою за зберігання. Позначимо через x_{ij} — кількість виробів, виготовлених фірмою в i-у місяці для доставки в j-му ($j \geqslant i$). Нехай c_{ij} — вартість таких виробів з врахуванням можливої оплати за зберігання. При $j < i$ приймемо $c_{ij} = M$, де M — доста-

тньо велике додатне число. Отож, отримали транспортну задачу з надлишком товару.

Перейдемо до розгляду симплекс-методу розв'язування стандартної транспортної задачі (1.28). Почнемо з побудови початкового базисного розв'язку. Передусім, зауважимо, що в задачі (1.28) mn змінних та $m+n$ обмежень-рівностей. У цьому разі сума перших m рівнянь дорівнює сумі n останніх. Отже, рівняння лінійно залежні. Покажемо, що матриця системи рівнянь має ранг $m+n-1$.

Для цього розглянемо загальну схему побудови *початкового базисного розв'язку* $x^{(1)}$. Вона полягає в послідовній побудові $m+n-1$ базисних компонент. Нехай $C=(c_{ij})_{m\times n}$ — матриця вартостей. Спочатку за деяким правилом вибирають пару (i_1,j_1) і першу компоненту приймаємо такою, що дорівнює $x^{(1)}_{i_1,j_1}=\min[s_{i_1},d_{j_1}]$. Якщо $s_{i_1}<d_{j_1}$, то припускаємо, що $x^{(1)}_{i_1,j_1}=s_{i_1}$, викреслюємо i_1-й рядок і перевизначається величина $d_{j_1}:=d_{j_1}-s_{i_1}$. Якщо $s_{i_1}>d_{j_1}$, то $x^{(1)}_{i_1,j_1}=d_{j_1}$ і викреслюється j_1-й стовпчик і перевизначається величина $s_{i_1}:=s_{i_1}-d_{j_1}$. Якщо $s_{i_1}=d_{j_1}$, то викреслюється або j_1-й стовпець, або i_1-рядок (але не одночасно). У цьому разі баланс зберігається і виникає транспортна задача меншої розмірності, в якій або $s_{i_1}=0$, або d_{j_1}.

У зведеній матриці знову за деяким правилом вибирається пара (i_2,j_2) і друга компонента $x^{(1)}_{i_2,j_2}$ припускається, що дорівнює $\min[s_{i_2},d_{j_2}]$ і т. д. Після чого залишиться матриця з одним рядком (або з одним стовпцем) розміром, наприклад, p. У ній викреслюють p разів одноелементні стовпці (або рядки). Зауважимо, що в 1×1 матриці одночасно викреслюють і рядок, і стовпець. Отже, загальне число викреслювань буде дорівнювати $m+n-1$ і внаслідок чого будуть визначені компоненти

$$x^{(1)}_{i_1j_1},\dots,x^{(1)}_{i_{m+n-1}j_{m+n-1}}.$$

Приймемо компоненти розв'язку $x^{(1)}$, які залишилися нульовими. За побудовою отримаємо, що розв'язок $x^{(1)}$ допустимий. Покажемо, що це базисний розв'язок з базисними компонентами

$$x^{(1)}_{ij},\ (i,j)\in I^{(1)}=\{(i_1,j_1),\dots,(i_{m+n-1},j_{m+n-1})\}.$$

Упорядкуємо змінні так:

$$x_{11}, x_{12}, \ldots, x_{1n}, x_{21}, \ldots, x_{2n}, \ldots, x_{m1}, \ldots, x_{mn}.$$

Розглянемо обмеження типу рівностей задачі (1.28). Розширена матриця системи рівностей набула вигляду

$$(A|b) = \left|\begin{array}{cccc|c} 1\ldots1 & & & 0 & s_1 \\ & 1\ldots1 & & & \vdots \\ 0 & & \ldots & & \vdots \\ & & & 1\ldots1 & s_m \\ \hline 1\ldots0 & 1\ldots0 & \ldots & 1\ldots0 & d_1 \\ \ddots & \ddots & & \ddots & \vdots \\ 0\ldots1 & 0\ldots1 & \ldots & 0\ldots1 & d_n \end{array}\right|$$

Позначимо через A_{ij} стовпець матриці A, який відповідає парі (i,j). Його i-та та $(m+j)$-та компоненти дорівніють 1, а решта дорівнюють нулю. Доведемо лінійну незалежність системи стовпців A_{ij}, $(i,j) \in I^{(1)}$. Покажемо, що відповідна цим стовпцям система лінійних рівнянь

$$\sum_{k=1}^{m+n-1} x_{i_k j_k} A_{i_k j_k} = b \tag{1.29}$$

має єдиний розв'язок. Насправді, якщо при побудові початкового БДР викреслений i_1-й рядок матриці C, тоді з i_1-го рівняння системи (1.22) компонента $x_{i_1 j_1}$ визначається однозначно і вона дорівнює $x^{(1)}_{i_1 j_1}$. Позначимо через $\hat{A}_{ij}, \hat{b}$, стовпці матриці $(A|b)$ після викреслювання i_1-го рядка. Після виключення з системи (1.22) змінної $x^{(1)}_{i_1 j_1}$ отримаємо аналогічну систему лінійних рівнянь

$$\sum_{k=2}^{m+n-1} x_{i_k j_k} \hat{A}_{i_k j_k} = \hat{b}.$$

З цієї системи знаходимо $x_{i_2 j_2} = x^{(2)}_{i_2 j_2}$. Аналогічно однозначно визначають й інші компоненти x_{ij} з номерами $(i,j) \in I^{(1)}$. Із

єдиності розв'язку системи рівнянь (1.29) призводить до незалежності стовпчиків її матриці і, отже, базисність розв'язку $x^{(1)}$. Звідси отримаємо, що ранг матриці A дорівнює $m+n-1$. Пари (i,j), які входять в множину $I^{(1)}$, будемо називати *базисними*.

Наведемо два методи побудови початкового БДР.

1. Метод "північно-західного кута". Приймемо $i_1 = 1$, $j_1 = 1$, тобто елемент, який стоїть в "північно-західному куті" матриці C. Після визначення компоненти $x_{11}^{(1)}$ і викреслювання першого рядка або першого стовпця в отриманій матриці беремо елемент в "північно-західному куті" і т. д. Зазвичай цей метод дає невдале початкове наближення, оскільки не враховується вартість c_{ij}.

2. Метод мінімального елемента. Нехай $c_{i_1 j_1}$ — мінімальний елемент матриці C. Після викреслювання в матриці C i_1-го рядка або j_1-го стовпця в отриманій матриці вибираємо елемент (i_2, j_2) з мінімальною вартістю і т. д.

Приклад 1.10. Розглянемо транспортну задачу, яка задана таблицею (рис. 1.2)

$$x^{(1)} = \begin{pmatrix} \underline{5} & 0 & 0 & 0 \\ \underline{2} & \underline{0} & 0 & 0 \\ 0 & \underline{1} & \underline{1} & \underline{1} \end{pmatrix}, \quad W(x^{(1)}) = 45$$

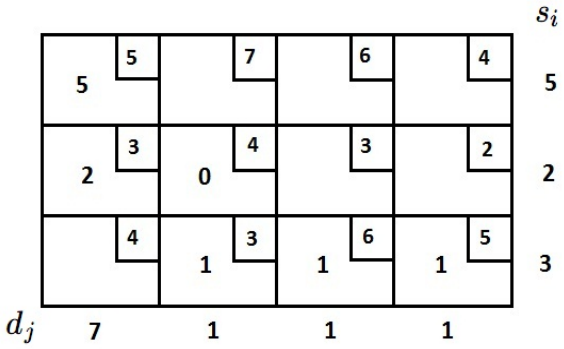


Рис. 1.2

У правих верхніх кутках клітинок зазначена вартість перевезення. У центрах клітинок розташовані базисні компоненти початкового розв'язку, який отримали методом "північно-західного кута". В матриці $x^{(1)}$ базисні елементи підкреслені. Якщо використати метод мінімального елемента, то

$$x^{(1)} = \begin{pmatrix} \underline{4} & 0 & \underline{1} & 0 \\ \underline{1} & 0 & 0 & \underline{1} \\ \underline{2} & \underline{1} & 0 & 0 \end{pmatrix}, \quad W(x^{(1)}) = 42.$$

Наведемо деякі властивості базисних компонентів довільного базисного розв'язку $x^{(1)}$. Нехай $I^{(1)} = \{(i_t, j_t), t = 1, \ldots, m+n-1\}$ — множина базисних пар, які відповідають $x^{(1)}$.

Лема 1.3. *Нехай $x^{(1)}$ — довільний БДР з множиною базисних пар $I^{(1)}$. Тоді правильні такі твердження:*

1) *у кожному рядку і в кожному стовпці матриці C міститься хоча б одна базисна пара;*
2) *будь-яка базисна пара не може бути одночасно єдиною в своєму рядку і в своєму стовпці матриці C;*
3) *знайдеться рядок (стовпець) матриці C, який містить єдину базисну пару.*

Уведемо такі бінарні відношення R_1 і R_2 на множині пар (i, j):

$$(i,j)R_1(i',j') \Leftrightarrow i = i'; \quad (i,j)R_2(i',j') \Leftrightarrow j = j'.$$

Очевидно, що відношення R_1 і R_2 транзитивні (перевірити!).

Розглядатимемо ланцюжки вигляду

$$(i_s, j_s)R_1(i_t, j_t)R_2(i_r, j_r)R_1 \ldots (i_l, j_l),$$

в яких відношення R_1 і R_2 чергуються і зв'язують базисні пари $(i_s, j_s), (i_t, j_t), (i_r, j_r), \ldots, (i_l, j_l)$. Наприклад, у базисному розв'язку

$$x^{(1)} = \begin{pmatrix} \underline{4} & 0 & \underline{1} & 0 \\ \underline{1} & 0 & 0 & \underline{1} \\ \underline{2} & \underline{1} & 0 & 0 \end{pmatrix}$$

ланцюжок $(2,4)R_1(2,1)R_2(3,1)R_1(3,2)$ зв'язує базисні пари $(2,4)$ і $(3,2)$.

Теорема 1.12. *Будь-які дві базисні пари (i_s, j_s) і (i_t, j_t), які відповідають базисним компонентам початкового базисного розв'язку, зв'язує єдиний ланцюжок.*

Доведення. Доведення проведемо індукцією за числом $m+n$, що будь-які дві базисні пари можна з'єднати ланцюжком. Для $m+n=3$, коли $m=2, n=1$ або $m=1, n=2$, обидві пари є базисними та з'єднаними ланцюжком. Припустимо, що це твердження правильне для всіх m і n таких, що $m+n \leqslant p$, де $p \geqslant 3$ — ціле число. Доведемо твердження при $m+n=p+1$. Використовуючи третє твердження леми 1.3, без обмеження загальності припустимо, що базисна пара $(i_1, j_1) = (1,1)$ єдина в першому стовпці.

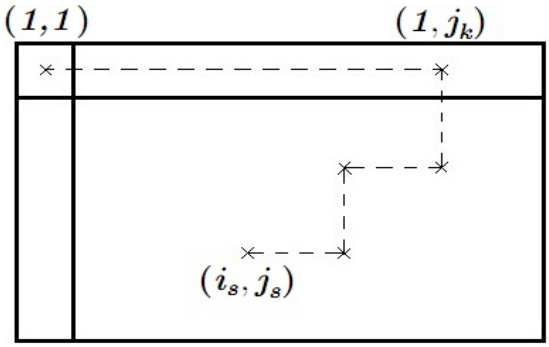


Рис. 1.3

Спочатку доведемо, що пара $(i_t, j_t) = (1,1)$ може бути з'єднана ланцюжком з парою (i_s, j_s). За другим твердженням леми 1.3 в першому рядку матриці C знайдеться базисна пара $(1, j_k) \neq (1,1)$. Якщо в матриці C викреслити перший стовпець, тоді в отриманій новій задачі пари $(i_k, j_k), k \neq t$, які залишилися, будуть базисні. За припущенням індукції пару $(1, j_k)$ можна з'єднати ланцюжком з парою (i_s, j_s). Тому і пару $(1,1)$ можна з'єднати ланцюжком з парою (i_s, j_s) (рис. 1.3).

Якщо пари (i_t, j_t) і (i_s, j_s) не збігаються з $(1,1)$, тоді їх можна з'єднати ланцюжком відповідно до припущення індукції.

Доведемо єдиність ланцюжка, що з'єднує дві базисні пари. Припустимо протилежне. Тоді існує цикл вигляду

$$(i_l, j_l)R_1(i_l, j_r)R_2 \dots R_1(i_p, j_l)R_2(i_l, j_l).$$

Звідси для базисних стовпчиків матриці A виконується рівність

$$A_{i_l,j_l} - A_{i_l,j_r} + \dots - A_{i_p,j_l} = 0,$$

що суперечить лінійній незалежності цих стовпців. □

Наслідок 1.13. *Якщо до базисних пар з $I^{(1)}$ додати не базисну пару (i,j), тоді існує єдиний цикл вигляду*

$$(i,j)R_1(i,j_l)R_2 \dots R_1(i_r,j)R_2(i,j),$$

де $(i,j_l)R_2 \dots R_1(i_r,j)$ — ланцюжок, який містить базисні пари.

Доведення. За лемою 1.3 в i-му рядку матриці C знайдеться базисна пара (i,j_t) і $(i,j)R_1(i,j_t)$. Аналогічно, в j-му стовпі матриці C знайдеться базисна пара (i_s,j) і $(i_s,j)R_2(i,j)$.

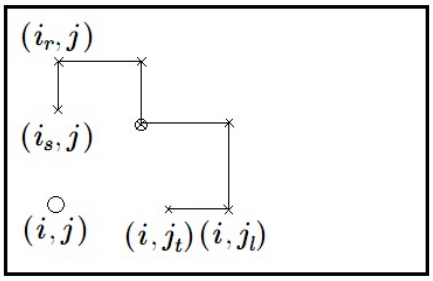


Рис. 1.4

Але пара (i,j_t) може бути з'єднана єдиним ланцюжком з парою (i_s,j) (рис. 1.4). Нехай цей ланцюжок набуває вигляду

$$(i,j_t)R_1(i,j_l)R_2 \dots R_1(i_r,j)R_2(i_s,j).$$

Тоді $(i,j)R_1(i,j_t)R_1(i,j_l)R_2 \dots R_1(i_r,j)R_2(i_s,j)R_2(i,j)$. За властивістю транзитивності відношення R_1 і R_2 отримуємо побудову шуканого циклу.

Як видно з доведення теореми 1.12, стовпець A_{ij} розкладається за базисними стовпцями $A_{ij_l}, \dots, A_{i_rj}$, які відповідають парам, що входять до циклу

$$A_{ij} = A_{ij_l} - \dots + A_{i_rj}.$$

З єдиності цього розвинення випливає єдиність циклу. □

Тепер перейдемо безпосередньо до методу розв'язування транспортної задачі (1.28). Задачу (1.28) можна записати як задачу максимізації функції

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

Почнемо симплекс-метод з початкового БДР $x^{(1)}$. Скористаємося конкретною структурою матриці обмеження. Зауважимо, що початкова симплекс-таблиця набула вигляду

W	c_{ij}	0
$x_{I^{(1)}}$	A	b

де $I^{(1)}$ — множина базисних пар розв'язку $x^{(1)}$. Зробимо так, щоб цільова функція W визначалася лише через небазисні змінні, для цього елементи нульового рядка симплекс-таблиці, що відповідають базисним стовпчикам, повинні бути нульовими. Складемо лінійну комбінацію рядків симплекс-таблиці, помноживши кожен i-й рядок на u_i і кожний $(m+j)$-й рядок на v_j. Отриману комбінацію віднімемо від нульового рядка, який у підсумку набуде вигляду

W	$c_{ij} - u_i - v_j$	$-\sum_{i=1}^{m} s_i u_i - \sum_{j=1}^{n} d_j v_j$

Для базисних пар $(i_t, j_t) \in I^{(1)}$ розглянемо систему рівнянь

$$u_{i_t} + v_{j_t} = c_{i_t j_t},\ t = 1, \ldots, n+m-1, \tag{1.23}$$

і виберемо величини $u_1, \ldots, u_m$, $v_1, \ldots, v_n$, які задовольняють цю систему. Тоді функція W буде визначена лише через небазисні змінні. Зауважимо, що система (1.23) містить $m+n-1$ рівнянь стосовно $m+n$ невідомих. Тому ця система має нескінченно багато розв'язків. Можна використовувати будь-який з цих розв'язків. Щоб виділити єдиний розв'язок, приймемо $u_{\tilde{j}} = 0$, де $(\tilde{i}, \tilde{j})$ — базисна пара, яка має найменшу вартість. У цьому разі $v_{\tilde{j}} = c_{\tilde{i}\tilde{j}}$. Покажемо, що значення решти змінних u_i, v_j визначають однозначно. Візьмемо будь-яку базисну пару $(i, j) \neq (\tilde{i}, \tilde{j})$. За теоремою 1.12 пари $(\tilde{i}, \tilde{j})$ і (i, j) можна з'єднати єдиним ланцюжком $(\tilde{i}, \tilde{j})R_1(\tilde{i}, j_l)R_2(i_k, j_l) \ldots R_1(i_r, j)R_2(i, j)$, рухаючись вздовж якого, можна знайти $v_{\tilde{j}}, u_{j_l}, \ldots, v_j$ і u_i. Оскільки в кожному рядку і стовпчці знайдеться хоча б одна базисна пара, то всі змінні u_i і v_j будуть однозначно визначені.

Початковий базисний розв'язок оптимальний, якщо для всіх не базисних пар виконується нерівність $c_{ij} - u_i - v_j \geqslant 0$. Нехай знайдеться пара (i, j), для якої виконана нерівність $c_{ij} - u_i - v_j < 0$. Тоді можна ввести в базис змінну x_{ij}. Знайдемо змінну, яка виводиться з базису. Розглянемо визначений у наслідку цикл

$$(i, j)R_1(i, j_l)R_2(i_k, j_l) \ldots R_1(i_r, j)R_2(i, j).$$

Побудуємо новий базисний розв'язок $x^{(2)}$. Для цього будемо збільшувати компоненту $x_{ij} \geqslant 0$ і збережемо при цьому всі обмеження задачі (1.21). Для цього достатньо на величину x_{ij} зменшити компоненту $x^{(1)}_{ij_l}$, збільшити на x_{ij} компоненту $x^{(1)}_{i_k j_l}$ і так далі, рухаючись вздовж циклу. У цьому разі вартість перевезень зміниться на величину

$$\begin{aligned} & x_{ij}\,(c_{ij} - c_{ij_l} + c_{i_k j_l} - \ldots - c_{i_r j}) \\ & = x_{ij}(c_{ij} - (u_i + v_{j_l}) + (u_{i_k} + v_{j_l}) - \ldots - (u_{i_r} + v_j)) \\ & = x_{ij}(c_{ij} - u_i - v_j), \end{aligned}$$

тобто зменшиться. Зрозуміло, що x_{ij} можна збільшувати до величини

$$x^{(1)}_{i^*j^*} = \min x^{(1)}_{i_tj_t},$$

де мінімум приймають за базисними парами циклу, які стоять на парних місцях. Покажемо, що при $x_{ij} = x^{(1)}_{i^*j^*}$ отримаємо новий базисний розв'язок $x^{(2)}$. У цьому випадку змінна $x_{i^*j^*}$ буде виведена з базису, а змінна x_{ij} — введена в нього. Справді, в розвинені вектора A_{ij} за базисними стовпцями

$$A_{ij} = A_{ij_l} - A_{i_kj_l} + \ldots + A_{i_rj}$$

вектор $A_{i^*j^*}$ входить з коефіцієнтом 1. Якщо в системі базисних стовпців $A_{i_tj_t}$, $t = 1, \ldots, m+n-1$, стовпець $A_{i^*j^*}$ замінити стовпцем A_{ij}, то отримаємо знову лінійно незалежну систему.

Для продовження симплекс-алгоритму треба знову розв'язати систему (1.23) для множини базисних пар $I^{(2)} = (I^{(1)} \setminus \{(i^*, j^*)\}) \cup \{(i, j)\}$ і перевірити умови оптимальності. Зауважимо, що величина $x^{(1)}_{i^*j^*}$ може дорівнювати нулю. Тоді $x^{(1)} = x^{(2)}$ і відбувається тільки зміна базису.

Наслідок 1.14. *Якщо величини s_i, d_j — цілі, тоді оптимальний базисний розв'язок, отриманий в наслідок роботи алгоритму, також буде цілочисловим.*

Справді, початковий базисний розв'язок — цілочисловий. Далі цілочисельність базисного розв'язку буде підтримуватися, оскільки в алгоритмі використовують лише операції додавання та віднімання.

Розв'яжемо методом потенціалів приклад 1.10 (рис. 1.5).

У центрах клітинок таблиці зазначені базисні компоненти початкового БДР $x^{(1)}$, отриманого методом мінімальної вартості. Цим компонентам відповідає система рівнянь

$$\begin{cases} u_1 + v_1 = 5; & u_1 + v_3 = 6; \\ u_2 + v_1 = 3; & u_2 + v_4 = 2; \quad u_2 = 0; \\ u_3 + v_1 = 4; & u_3 + v_2 = 3. \end{cases}$$

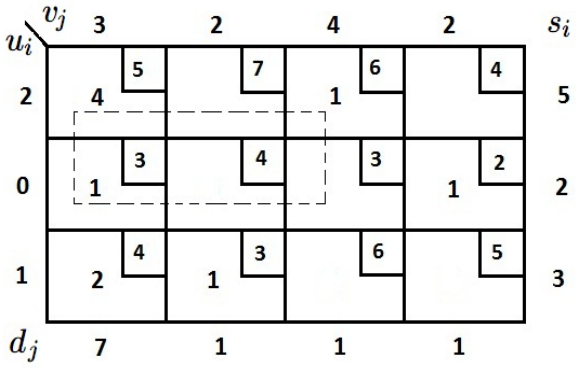


Рис. 1.5

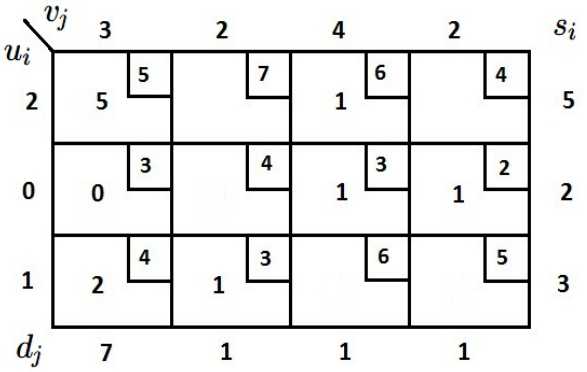


Рис. 1.6

Розв'язок системи показано зліва і зверху від таблиці. Умова оптимальності не виконується для пари (2,3): $c_{23} - u_2 - v_3 = -1 < 0$. Знаходимо цикл, який з'єднює пару (2,3) з базисними парами

$$(2,3)R_1(2,1)R_2(1,1)R_1(1,3)R_2(2,3).$$

На його парних місцях компоненти розв'язку $x^{(1)}$ дорівнюють 1. Тому змінну x_{23} можна збільшити на 1. Звідси отримаємо таблицю (рис. 1.6).

Тут змінна x_{13} виведена з базису, а змінна x_{23} введена в нього. Розв'язуємо систему рівнянь

$$\begin{cases} u_1 + v_1 = 5; & u_2 + v_3 = 3; \\ u_2 + v_1 = 3; & u_2 + v_4 = 2; \quad u_2 = 0; \\ u_3 + v_1 = 4; & u_3 + v_2 = 3. \end{cases}$$

Умова оптимальності виконана і $x^{(2)}$ — оптимальний розв'язок задачі вартості $W(x^{(2)}) = 41$.

Задачі

1.7 У прикладі 1.10 знайти всі оптимальні базисні розв'язки транспортної задачі.

Відповідь. Множиною всіх розв'язків транспортної задачі є відрізок $[x^{(2)}, x^{(3)}]$ у просторі 3×4-матриць, де

$$x^{(3)} = \begin{pmatrix} 4 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 2 & 1 & 0 & 0 \end{pmatrix}.$$

1.8 Розв'язати задачу про призначення з матрицею витрат

$$C = (c_{ij}) = \begin{pmatrix} 9 & 1 & 5 & 6 \\ 7 & 2 & 8 & 8 \\ 6 & 8 & 3 & 7 \\ 1 & 7 & 4 & 5 \end{pmatrix},$$

звівши її до транспортної задачі. Знайти всі оптимальні розв'язки.

Відповідь. Єдиний оптимальний розв'язок

$$\begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}.$$

1.9 На чотирьох заводах виготовляють деталі для станків, які потім відправляють у три фірми. Вартість перевезень визначається матрицею

$$C = (c_{ij}) = \begin{pmatrix} 1 & 3 & 2 \\ 3 & 5 & 8 \\ 2 & 7 & 4 \\ 5 & 3 & 6 \end{pmatrix},$$

а потужності і потреби, відповідно $s = (25, 10, 30, 20)$ і $d = (30, 40, 15)$. Потрібно спланувати розподіл деталей між фірмами так, щоб вартість перевезень була найменшою.

Відповідь. $W_{\min} = 230$.

Розділ 2

Цілочислове програмування

2.1 Формулювання задачі цілочислового LP

Розглянемо цілочислову задачу LP

$$\begin{aligned} &\max W(x) = \sum_{j=1}^{n} c_j x_j \\ &\text{за умов} \qquad \sum_{j=1}^{n} a_{ij} x_j \leqslant b_i,\ i = 1, \ldots, m, \\ &\qquad\qquad x_j \geqslant 0,\ x_i \in \mathbb{Z},\ j = 1, \ldots, n\,, \end{aligned} \tag{2.1}$$

де $\mathbb{Z}$ — множина цілих чисел.

Якщо $x_j = \{0, 1\}$, то задачу (2.1) будемо називати *задачею булевого лінійного програмування*. Множину допустимих розв'язків задачі (2.1) позначимо через X_0. Якщо відкинути умову цілочисельності, то отримаємо "неперервну" задачу (1.15) з множиною допустимих розв'язків $X \supset X_0$.

Наявні приклади задач цілочислового програмування, в яких оптимальний розв'язок можна отриманити як розв'язок відповідної неперервної задачі (1.15). Наприклад, у транспортній задачі (1.28) з виробництвом штучного продукту величини s_j, d_j –

цілі, а розв'язок задачі методом потенціалів призводить до оптимального цілочислового розв'язку.

Іноді вважають, що для розв'язання задачі (2.1) треба розв'язати неперервну задачу (1.15) і округлити до найближчої цілої компоненти отриманого оптимального розв'язку. Проте цей метод не завжди приводить до мети.

Приклад 2.1. Розглянемо задачу

$$\begin{aligned} &\max W = x_1 + x_2 \\ \text{за умов} \quad &\Big(1 + \frac{1}{a}\Big)x_1 + 1 \leqslant x_2 \leqslant \Big(1 - \frac{1}{2a}\Big)x_1 + \frac{3}{2}, \\ &x_j \geqslant 0,\ x_j \in \mathbb{Z},\ j = 1, 2, \end{aligned}$$

де $a > 0$ — параметр.

Покажемо, що $x^0 = (0, 1)$ — єдиний допустимий розв'язок. Справді, припустимо, що $(x_1, x_2) \in X_0$. Нехай $x_1 > 0$. З першої нерівності обмеження одержимо, що $x_2 - x_1 \geqslant 2$, а з другої — $x_2 - x_1 \leqslant 1$ (суперечність). Нехай $x_1 = 0$. Тоді $1 \leqslant x_2 \leqslant 3/2$ і $x_2 = 1$. Отже, $x^0 = (0, 1)$ — оптимальний розв'язок задачі (2.1) і $W(x^*) = 1$. Для оптимального розв'язку x^* відповідної неперервної задачі нерівності обмеження перетворюються в рівності і $x^* = (a/3, (a+4)/3)$. При $a > 3$ заокруглення x^* до найближчого цілого не призводить до розв'язку цілочислової задачі (розв'язки не є допустимими). При $a \to \infty$ компоненти розв'язку x^* прямують до нескінченності. ▲

2.2 Метод Гоморі

У цьому параграфі припустимо, що всі коефіцієнти a_{ij}, c_j, b_i задачі (2.1) цілі числа. Головні обмеження в ній можна задати у формі рівностей, увівши слабкі змінні x_j, $j = n+1, \ldots, n+m$. Для наведеного припущення, якщо $x \in X_0$, то відповідні значення слабких змінних також будуть цілими.

Метод Гоморі розв'язання задачі (2.1) побудований на використанні симплекс-методу та послідовному введенні додаткових обмежень (*зрізу Гоморі*), які не змінюють множину X_0.

Нехай a, b — дійсні числа, c — ціле число.

Означення 2.1. *Кажуть, що числа a і b можна порівняти за модулем c (позначення: $a \equiv b(\mod c)$), якщо різниця $(a-b)$ — ціле число, яке ділиться без залишку на c. Якщо буде порівняння за модулем 1, то запис $(\mod 1)$ опускатимемо.*

Приклади порівнянь: $9 \equiv 3(\mod 2)$, $3/2 \equiv 5/2$, $a \equiv 0$ для будь-якого цілого a.

Порівняння за модулем одного числа можна додавати і віднімати як рівняння, а також множити на цілі числа.

Схема побудови зрізу Гоморі

Припустимо, що для розв'язування неперервної задачі LP (1.15) симплекс-методом на деякому кроці отримано поточну МБН $I = \{B(1), \ldots, B(m)\}$. Нехай

$$x_{B(i)} + \sum_{j \notin J} a'_{ij} x_j = b'_i \,, \tag{2.2}$$

— рівняння, яке відповідає i-му рядку симплекс-таблиці, в якій число b'_i нецiле. Цей рядок будемо називати опорним. Звідси випливає порівняння

$$x_{B(i)} + \sum_{j \notin I} a'_{ij} x_j \equiv b'_i \,. \tag{2.3}$$

Оскільки $x_j \in \mathbb{Z}$, $x_j \equiv 0$, $j = 1, \ldots, n+m$, і $0 \equiv 1$, то з (2.3) випливає

$$\sum_{j \notin I} f_{ij} x_j \equiv f_i \,, \tag{2.4}$$

де $f_{ij} = a'_{ij} - [a'_{ij}]$; $f_i = b'_i - [b'_i]$ — дробові частини чисел a'_{ij} і b'_i.

Розглянемо оптимізаційну задачу

$$\begin{aligned} & \min \sum_{j \notin I} f_{ij} x_j \\ \text{за умов} \quad & \sum_{j \notin I} f_{ij} x_j \equiv f_i, \\ & x_j \in \mathbb{Z},\ x_j \geqslant 0,\ j \notin I. \end{aligned} \tag{2.5}$$

Очевидно, що мінімальне значення цільової функції задачі (2.5) дорівнює $f_i + k$, де k – деяке ціле число. Зауважимо, що $k \geqslant 0$, оскільки $0 < f_i < 1$. Для будь-якого розв'язку $x \in X_0$ правильна нерівність

$$\sum_{j \notin I} f_{ij} x_j \geqslant f_i + k\,, \tag{2.6}$$

яку будемо називати *модифікованим зрізом Гоморі*. Класичний зріз Гоморі відповідає слабшій нерівності (2.6) при $k = 0$. Зазначимо, що за опорний рядок можна взяти і нульовий рядок симплекс-таблиці.

Приклад 2.2. Нехай опорний рядок симплекс-таблиці визначається рівнянням

$$x_2 + \frac{17}{14} x_3 - \frac{9}{14} x_4 = \frac{22}{7}\,.$$

Звідси випливає порівняння

$$x_2 + \frac{17}{14} x_3 - \frac{9}{14} x_4 \equiv \frac{22}{7} \implies \frac{3}{14} x_3 + \frac{5}{14} x_4 \equiv \frac{1}{7}\,.$$

Розв'яжемо тепер таку оптимізаційну задачу:

$$\begin{aligned} &\min \ \frac{3}{14} x_3 + \frac{5}{14} x_4, \\ &\text{за умов} \quad \frac{3}{14} x_3 + \frac{5}{14} x_4 \equiv \frac{1}{7},\ \ x_3, x_4 \geqslant 0,\ \ x_3, x_4 \in \mathbb{Z}. \end{aligned}$$

Перебираючи значення x_3, $x_4 = 0, 1, 2, \ldots$, знаходимо, що мінімум досягається при $\hat{x_3} = \hat{x_4} = 2$ і дорівнює $8/7$. Отже, для будь-якого розв'язку $x \in X_0$ правильна нерівність

$$\frac{3}{14} x_3 + \frac{5}{14} x_4 \geqslant \frac{8}{7}\,,$$

яка є модифікованим зрізом Гоморі. ▲

Наведемо тепер **алгоритм Гоморі** розв'язання задачі (2.1), який також називається **дробовим двоїстим алгоритмом**. Метод Гоморі починається з пошуку симплекс-методом оптимального розв'язку неперервної задачі (1.15)

$$\max_{x\in X} W(x) = W(x^1)\,.$$

Після цього отримана симплекс-таблиця перетворюється в строго двоїсто допустиму двоїстим симплекс-методом. Далі проглядається останній рядок симплекс-таблиці, починаючи з нульової компоненти. Перший рядок, який містить в останньому стовпці неціле число, береться за опорний. Будується (модифікований) зріз Гоморі, який записують у вигляді рівності з використанням слабкої змінної s_1 і додають як останній рядок симплекс-таблиці. Додають також базисний стовпець $e_{m+2} \in \mathbb{R}^{m+2}$, який відповідає змінній s_1.

В алгоритмі Гоморі виділяють так звані великі та малі ітерації. Перша велика ітерація фактично вже описана. Наступні великі ітерації включають операції заміщення за двоїстим симплекс-алгоритмом, який називають малою ітерацією. Вибір опорного стовпця в малій ітерації описаний в теоремі 1.9. Розглянемо як відбувається t велика ітерація, $t \geqslant 2$. До цього моменту формується неперервна задача $(1.15)_t$, що є задачею (1.15) з $t-1$ зрізами Гоморі. Які записані у вигляді рівностей з використанням слабких змінних $s_1, \ldots, s_{t-1}$. Симплекс-таблиця задачі $(1.15)_t$ строго двоїсто допустима. Велика t операція починається з малої ітерації, в якій опорний останній рядок симплекс-таблиці відповідає $(t-1)$-у зрізу Гоморі. У цьому разі змінна s_{t-1} виводиться з базису. Наступні малі ітерації проводять за такими правилами:

1) за опорний вибирають i-й рядок, який містить довільну поточну базисну компоненту $x_{B(i)} = b'_i < 0$, де $B(i) \in \{1, \ldots, n+m\}$;

2) малі ітерації припиняються, якщо в поточному базисному розв'язку

$$x_j \geqslant 0,\ j = 1, \ldots, n+m;$$

3) алгоритм Гоморі припиняє роботу, якщо в процесі виконання малих ітерацій з'ясовується, що цільова функція задачі, двоїстої до $(1.15)_t$, не обмежена знизу.

Виконання умови 3) означає, що додавання $(t-1)$-го зрізу Гоморі робить несумісними обмеження задачі $(1.15)_t$. Отже, в цьому випадку множина X_0 — допустимих розв'язків задачі (2.1) порожня. Зазначимо, що згідно з теоремою 1.9 велика ітерація може містити лише скінчене число малих ітерацій.

Після завершення малих ітерацій проглядається останній стовпець симплекс-таблиці, починаючи з нульовою рядка. Якщо всі його компоненти – цілі, то поточний базисний розв'язок визначає оптимальний розв'язок $x^t \in X_0$ задачі (2.1) (див. далі лему 2.1). В іншому випадку будується наступний зріз Гоморі, який записують у вигляді рівності з використанням слабкої змінної s_t, до симплекс-таблиці додається рядок і базисний стовпець, що відповідає змінній s_t. На цьому t велика ітерація закінчується.

У сформульованому алгоритмі з кожною великою ітерацією зростає кількість рядків і стовпців симплекс-таблиці, що є його недоліком. Крім того, після завершення малих ітерацій симплекс-таблиця може не бути прямо допустимою, тобто деякі базисні змінні s_j від'ємні й умова оптимальності для розв'язку x^t в задачі $(1.15)_t$ порушується. У зв'язку з цим потрібне деяке уточнення алгоритму.

На початку $(t+1)$-ї великої ітерації змінна s_t виводиться з базису, але при подальших малих ітераціях може бути повернута в базис. Покажемо, що далі змінна s_t залишиться в базисі. Справді, для того, щоб змінна s_t була повторно виведена з базису, потрібно, щоб i-й рядок, який її містить, був опорним, тобто $s_t = b'_i < 0$, що суперечить правилу 1 алгоритму. Отож, рядок симплекс-таблиці, яка містить повторно введену в базис змінну s_t, ніколи не буде опорним. Цей рядок не впливає на подальші обчислення і її можна викреслити. Після цього в симплекс-таблиці утворюється принаймні один нульовий стовпець, який відповідає базисній змінній s_t. Можуть з'явитися й інші нульові стовпці, що відповідають деяким іншим не базисним змінним

x_j, $j \in I_1$. Можна прийняти $x_j = 0$, $j \in I_1$, до кінця роботи алгоритму, що не вплине на значення цільової функції задачі $(1.15)_t$. Треба викреслити всі нульові стовпці. Згідно з теоремою 1.9 симплекс-таблиця залишиться строго двоїсто допустимою. Тепер після закінчення малих ітерацій змінні s_j не можуть бути базисними. Внаслідок цих змін в алгоритмі Гоморі отримуємо таке твердження.

Лема 2.1. *У процесі виконання алгоритму Гоморі число рядків симплекс-таблиці не перевершує $n + m$, а після завершення в t-й великій ітерації малих ітерацій x^t — оптимальний розв'язок задачі $(1.15)_t$.*

Приклад 2.3. Розв'яжемо методом Гоморі задачу цілочислового LP

$$\max W = 2x_1 + 3x_2$$
$$\text{за умов} \qquad x_1 + 3x_2 \leqslant 9,\ 4x_1 + x_2 \leqslant 13,\ \ x_1, x_2 \geqslant 0,\ x_1, x_2 \in \mathbb{Z}.$$

Введемо слабкі змінні x_3, x_4 і розв'яжемо неперервну задачу (1.15) симплекс-методом. Обчислення наведені в таких трьох симплекс-таблицях:

W	-2	-3	0	0	0
x_3	1	$\underline{3}$	1	0	9
x_4	4	1	0	1	13

,

W	-1	0	1	0	9
x_2	1/3	1	1/3	0	3
x_4	$\underline{11/3}$	0	$-1/3$	1	10

,

W	0	0	10/11	3/11	129/11
x_2	0	1	4/11	$-1/11$	23/11
x_1	1	0	$-1/11$	3/11	30/11

.

Оптимальний розв'язок неперервної задачі $x^1 = (30/11, 23/11)$ містить не цілі компоненти. Візьмемо рівняння

$$W + \frac{10}{11}x_3 + \frac{3}{11}x_4 = \frac{129}{11},$$

яке відповідає нульовому рядку, і побудуємо зріз Гоморі. Отримаємо

$$\frac{10}{11}x_3 + \frac{3}{11}x_4 \equiv \frac{8}{11}\,.$$

Звідси знаходимо модифікований зріз: $10x_3/11+3x_4/11 \geqslant 19/11$.

Увівши слабку змінну s_1, останню нерівність запишемо як рівність $-10x_3/11-3x_4/11+s_1 = -19/11$ і додамо його в симплекс-таблицю

W	0	0	10/11	3/11	0	129/11
x_2	0	1	4/11	−1/11	0	23/11
x_1	1	0	−1/11	3/11	0	30/11
s_1	0	0	−10/11	−3/11	1	−19/11

.

Розв'яжемо неперервну задачу $(1.15)_2$ з додатковим обмеженням, використовуючи двоїстий симплекс-алгоритм. Тут опорним є четвертий стовпець, оскільки $\overline{A}_4/a_{34} \succ \overline{A}_3/a_{33}$. Після операції заміщення отримаємо

W	0	0	0	0	1	10
x_2	0	1	2/3	0	−1/3	8/3
x_1	1	0	−1	0	1	1
x_4	0	0	10/3	1	−11/3	19/3

.

Тут $x^2 = (1, 8/3)$ — оптимальний розв'язок задачі $(1.15)_2$. За першим опорним рядком будуємо другий зріз: $2x_3/3 + 2s_1/32/3$. Додамо його в симплекс-таблицю, ввівши слабку змінну s_2

W	0	0	0	0	1	0	10
x_2	0	1	2/3	0	−1/3	0	8/3
x_1	1	0	−1	0	1	0	1
x_4	0	0	10/3	1	−11/3	0	19/10
s_2	0	0	−2/3	0	−2/3	1	−2/3

.

Після перетворення вона набуває такого вигляду:

W	0	0	0	0	1	0	10
x_2	0	1	0	0	−1	1	2
x_1	1	0	0	0	2	−3/2	2
x_4	0	0	0	1	−7	5	3
x_3	0	0	1	0	1	−3/2	1

.

Звідси отримуємо оптимальний розв'язок $x^3 = (2, 2)$ задачі (2.1). Якщо використовувати класичні зрізи Гоморі, то їх потрібно було б чотири замість двох модифікованих (перевірте). ▲

Тепер розглянемо питання про збіжність методу Гоморі.

Теорема 2.1. *Нехай X множина допустимих розв'язків задачі (1.15) обмежена. Тоді алгоритм Гоморі розв'язання задачі (2.1) збігається за скінченну кількість ітерацій.*

Доведення. Позначимо через J^t – МБН, а через

$$b_i^t,\ a_{ij}^t,\ i = 0, 1, \ldots, m+n,\ j = 1, \ldots, m+n+n_t\,,$$

– компоненти стовпців симплекс-таблиці, що утворилася безпосередньо після завершення великої ітерації t, де n_t – ціле число, яке не перевершує n. Зазначимо, що b_0^t – оптимальне значення цільової функції задачі $(1.15)_t$ (і в двоїстій до неї).

Припустимо, що для деякої задачі (2.1) алгоритм Гоморі виконує нескінченну кількість ітерацій. Тоді в кожній великій ітерації t множина допустимих розв'язків задачі $(1.15)_t$ не порожня. Тому цільова функція задачі, двоїстої до $(1.15)_t$, обмежена знизу константою $L = \min\limits_{x \in X} W(x)$. Звідси випливає, що величини b_0^t, $t \geqslant 1$, обмежені знизу константою L. При кожному t останній стовпець $b^t = (b_0^t, b_1^t, \ldots, b_{m+n_t}^t)^T$ невід'ємний і містить не цілі компоненти. Оскільки симплекс-таблиця в ході алгоритму залишається строго двоїсто допустимою, то за теоремою 1.9 останній рядок симплекс-таблиці лексикографічно спадає. Отже, перша ненульова різниця $b_i^{t+1} - b_i^t$, $i = 0, 1, \ldots, m+n_t$ від'ємна.

Нехай b_0^t — неціле число. Тоді нульовий рядок симплекс-таблиці буде опорним і в неперервній задачі $(1.15)_{t+1}$ з'явиться обмеження

$$-\sum_{j \notin J^t} f_{0j}^t x_j + s_t = -(f_0^t + k_0^t)\,,$$

де $f_{0j}^t = a_{0j}^t - [f_{0j}^t]$, $j \notin J^t$; $f_0^t = b_0^t - [b_0^t]$; k_0^t — ціле невід'ємне число. Зауважимо, що у випадку використання стандартного зрізу Гоморі $k_0^t = 0$.

Нехай p-й стовпець — опорний. Тоді $f^t_{0p} > 0$ і $a^t_{0p} = [a^t_{0p}] + f^t_{0p} > 0$, оскільки p-й стовпець лексикографічно додатний. Після операції заміщення величина b^t_0 перетворюється на величину

$$b^{t+1}_0 = b^t_0 - (f^t_0 + k^t_0)\frac{a^t_{0p}}{f^t_{0p}} \leqslant b^t_0 - (f^t_0 + k^t_0)\frac{f^t_{0p}}{f^t_{0p}} = [b^t_0] - k^t_0 \leqslant [b^t_0]\,.$$

Остання нерівність означає, що величина b^t_0 зменшується принаймні до найближчого цілого. Оскільки величина b^t_0 обмежена знизу константою L, то її дробове значення може траплятися тільки скінченну кількість разів. Тому починаючи з деякого t_1, b^t_0 — ціле число, яке не залежить від t.

Нехай при деякому $t \geqslant t_1$ величина b^t_1 — нецілe число. Тоді $b^t_1 > 0$ і в неперервній задачі $(1.15)_{t+1}$ з'явиться останнє обмеження

$$-\sum_{j \notin B^t} f^t_{1j} x_j + s_t = -(f^t_1 + k^t_1),$$

де $f^t_{1j} = a^t_{1j} - [f^t_{1j}]$, $j \notin B^t$, $f^t_1 = b^t_1 - [b^t_1]$, k^t_1 — ціле невід'ємне число.

Покажемо, що в нульовому рядку симплекс-таблиці $a^t_{0p} = 0$. Справді, в іншому випадку $a^t_{0p} > 0$ і за лексикографічним правилом вибору опорного стовпця (див. теорему 1.9) $a^t_{0j} > 0$ для всіх j, таких що $f^t_{1j} > 0$. Додаючи до нульового рядка симплекс-таблиці її останній рядок, помножений на мале ціле додатне число, можна домогтися зменшення значення цільової функції задачі, двоїстої до $(1.15)_{t+1}$(протиріччя з визначенням b^0_t). Отож, $a^t_{0p} = 0$. Тому $a^t_{1p} \geqslant 0$. Але з нерівності $f^t_{1p} > 0$ випливає, що $a^t_{1p} = [a^t_{1p}] + f^t_{1p} > 0$. Тому після операції заміщення величина b^t_1 зменшується принаймні до найближчого цілого. Оскільки $b^t_1 \geqslant 0$ дробове значення b^t_1 може траплятися тільки скінченну кількість разів. Звідси, починаючи з деякого t_2, величина b^t_1 стає цілою і сталою. За лемою 2.1 кількість рядків симплекс-таблиці не перевищує $m + n$. Продовживши аналогічні міркування, можна показати, що, починаючи з деякого t_{m+n}, всі компоненти стовпця b^t будуть цілими і сталими, що означає оптимальність розв'язку x^t для задачі (2.1) Отже, алгоритм Гоморі не може виконувати нескінченну кількість ітерацій, що суперечить припущенню. $\square$

При використанні модифікованих зрізів потрібно багато разів розв'язувати задачу (2.5). Покажемо, що для її розв'язування можна використовувати метод ефективніший, ніж алгоритм перебору. Обмежимося випадком двох змінних і запишемо задачу (2.5) у вигляді

$$\min W = \frac{a_1}{r}x_1 + \frac{a_2}{r}x_2,$$
$$\text{за умов} \qquad \frac{a_1}{r}x_1 + \frac{a_2}{r}x_2 \equiv \frac{b}{r},\ x_1, x_2 \geqslant 0,\ x_1, x_2 \in \mathbb{Z},$$

або

$$\begin{aligned} &\min\ W = \mu, \\ &\text{за умов} \quad a_1x_1 + a_2x_2 = b + \mu r, \\ &\qquad\qquad x_1, x_2, \mu \geqslant 0,\ x_1, x_2, \mu \in \mathbb{Z}, \end{aligned} \tag{2.7}$$

де a_1, a_2, r — додатні цілі числа; $\max[a_1, a_2, b] < r$. Припускаємо, що обмеження задачі (2.2) сумісні.

Через НСД$(h, l, \ldots, p)$ позначимо найбільший спільний дільник цілих чисел $h, l, \ldots, p$. Без втрати загальності будемо вважати, що НСД$(a_1, a_2, b) = 1$. Приймемо НСД$(a_1, a_2) = d$. Розглянемо два випадки.

1. Нехай $d = 1$. Загальний розв'язок рівняння $a_1x_1 + a_2x_2 = b + \mu r$ в цілих числах записують у вигляді

$$x_1(t) = ta_2 + (b + \mu r)x_1^0,\ \ x_2(t) = -ta_1 + (b + \mu r)x_2^0\,,$$

де (x_1^0, x_2^0) — частковий розв'язок рівняння $a_1x_1 + a_2x_2 = 1$, а параметр t набуде будь-яких цілих значень. Приймемо

$$\psi_1(\mu) = \frac{(b + \mu r)x_2^0}{a_1},\ \ \psi_2(\mu) = -\frac{(b + \mu r)x_1^0}{a_2}\,.$$

При заданому значенні μ розв'язок $(x_1(t), x_2(t))$ невід'ємний тільки у випадку, коли $t \in [\psi_2(\mu), \psi_1(\mu)]$. Звідси для існування невід'ємного цілочислового розв'язку необхідно і достатньо, щоб $[\psi_1(\mu)] \geqslant \psi_2(\mu)$.

Тому знаходимо мінімальне ціле μ^0, яке задовольняє цю нерівність. Воно буде відповідати мінімальному значенню μ в задачі (2.2). Зауважимо, що ціле μ, яке задовольняє нерівність $[\psi_1(\mu)] \geqslant \psi_2(\mu)$, завжди знайдеться, оскільки різниця

$$\psi_1(\mu) - \psi_2(\mu) = (b + \mu r)\left(\frac{x_1^0}{a_2} + \frac{x_2^0}{a_1}\right) = \frac{b + \mu r}{a_1 a_2}$$

зі збільшенням μ необмежено збільшується. У підсумку отримуємо зріз

$$\frac{a_1}{r}x_1 + \frac{a_2}{r}x_2\frac{b}{r} + \mu^0 .$$

2. Нехай $d > 1$. Тоді числа a_i, $i = 1, 2$, зобразимо у вигляді $a_i = a_i' d$, де $a_i' > 0$, $a_i' \in \mathbb{Z}$, $i = 1, 2$, НСД$(a_1', a_2') = 1$. У цьому випадку рівняння $a_1 x_1 + a_2 x_2 = b$ не має розв'язку в цілих числах. Покажемо, що d і r – взаємно прості числа, тобто НСД$(d, r) = 1$. Справді, припустимо, що НСД$(d, r) = d_1 > 1$. Тоді з рівняння $a_1 x_1 + a_2 x_2 = b + \mu r$ випливає, що d_1 ділить b, але це суперечить рівності НСД$(a_1, a_2, b) = 1$.

Розглянемо рівняння $kd - \mu r = b$ стосовно цілих змінних k і μ. Це рівняння має розв'язки в цілих числах, оскільки НСД$(d, r) = 1$. Візьмемо розв'язок k^0, μ^0, який задовольняє умову $0 < k^0 < r$. Підставивши $a_1 = a_2' d$, $a_2 = a_2' d$, $b = k^0 d - \mu^0 r$ в рівняння $a_1 x_1 + a_2 x_2 = b + \mu r$, отримаємо

$$a_1' d x_1 + a_2' d x_2 = k^0 d + (\mu - \mu^0) r .$$

З нього випливає, що d ділить $\mu - \mu^0$, тобто $\mu - \mu^0 = sd$. Після скорочення на d, отримуємо $a_1' x_1 + a_2' x_2 = k^0 + sr$ і, як у випадку 1, знаходимо мінімальне s^0, яке відповідне оптимальному розв'язку здачі

$$\begin{aligned} &\min W = s, \\ &\text{за умов} \quad a_1' x_1 + a_2' x_2 = k^0 + sr, \\ &\qquad\qquad x_1, x_2, s \geqslant 0,\ x_1, x_2, s \in \mathbb{Z}. \end{aligned}$$

Звідси знаходимо зріз

$$\frac{a_1'}{r}x_1 + \frac{a_2'}{r}x_2 \geqslant \frac{k^0}{r} + s\,.$$

Приклад 2.4. Знайдемо модифікований зріз Гоморі для порівняння

$$\frac{710}{401}x_1 + \frac{158}{401}x_2 \equiv \frac{1}{401}\,.$$

Тут $a_1 = 710$, $a_2 = 158$, $b = 1$, $r = 401$, $d = (710, 158) = 2$. Рівняння $dk - r\mu = b$ має розв'язок $(k^0, \mu^0) = (201, 1)$.

Далі $a_1' = 355$, $a_2' = 79$. Рівняння $a_1'x_1 + a_2'x_2 = 1$ має частковий розв'язок $(x_1^0, x_2^0) = (77, -346)$. Звідси

$$\psi_1(s) = \frac{(k^0 + rs)x_2^0}{a_1'} = -\frac{(201 + 401s)346}{155},$$

$$\psi_2(s) = -\frac{(k^0 + rs)x_1^0}{a_2'} = -\frac{(201 + 401s)77}{79}.$$

Мінімальне $s \geqslant 0$, яке задовольняє нерівність $[\psi_1(s)] \geqslant \psi_2(s)$, дорівнює 6. У цьому разі $\psi_1(6) \approx -2540,9$, $\psi_2(6) = -2541$. Отримаємо зріз

$$\frac{155}{401}x_1 + \frac{79}{401}x_2 \geqslant \frac{201}{401} + 6 = \frac{2406}{401}\,.$$

▲

Задачі

2.1. При обмеженнях з прикладу 2.3 розв'яжіть задачу цілочислового LP для таких цільових функцій:

1) $W = 3x_1 + x_2$;

2) $W = x_1 + 3x_2$.

Відповідь. 1) $x^* = (3, 1)$; 2) $x^* = (0, 3)$.

2.2. Знайти розв'язок задачі

$$\begin{aligned} \max\, W(x) =& 6x_1 + 5x_2, \\ \text{за умов} \quad & 5x_1 + 4x_2 \leqslant 20, \\ & 4x_1 + 7x_2 \leqslant 28, \\ & 0 \leqslant x_1 \leqslant 3,\ x_2 \geqslant 0,\ x_1, x_2 \in \mathbb{Z}. \end{aligned}$$

Відповідь. $x^* = (3, 1)$, $W_{\max} = 23$.

2.3 Метод гілок і меж

Спочатку викладемо загальну схему методу, а потім його конкретну реалізацію для задачі цілочислового LP.

Нехай цільова функція $W(x)$ визначена на скінченній множині розв'язків X_0. Розглянемо задачу

$$\max_{x \in X_0} W(x) = W(x^0)\,. \tag{2.8}$$

Вихідну множину X_0 можна помістити в ширшу множину $X \supset X_0$. Множина X може містити скінченну або нескінченну множину розв'язків.

Припустимо, що ми вміємо розбивати (розгалужувати) множину X на такі підмножини A, для яких підзадача $\max\limits_{x \in A} W(x) = W(x_A) = W_A$ розв'язується досить нескладно. Якщо оптимальний розв'язок x_A підзадачі допустимий (тобто $x_A \in X_0$), то величина W_A є оцінкою (межа) знизу для шуканого максимуму $W(x_0)$.

Розглянемо метод гілок і меж для задачі (2.8) детальніше. Нехай N — поточна оцінка знизу для максимуму $W(x_0)$ задачі (2.8). Ця оцінка або дорівнює $-\infty$, або досягається для деякого поточного допустимого розв'язку. Спочатку припустимо, що $N = -\infty$ і розв'язуємо задачу

$$\max_{x \in X} W(x) = W(x^*) = W_X.$$

Якщо x^* належить X_0, то x^* — оптимальний розв'язок задачі (2.8), оскільки $X_0 \subset X$.

Припустимо, що $x^* \notin X_0$. У цьому випадку множину X розбиваємо на підмножини $X_1, \ldots, X_m$, послідовно розв'язуємо підзадачі і знаходимо величини $W_{X_i} = W(x_{X_i})$, $i = 1, \ldots, m$, зокрема за такими правилами відкидаємо підмножини X_i:

1) $N \geqslant W_{X_i}$. В цьому випадку N – скінченна величина, яка з побудови досягається на поточному допустимому розв'язку задачі (2.8). Тому всі розв'язки з множини X_i мають ненайліпші значення цільової функції і можуть бути відкинуті;

2) $N < W_{X_i}$ і $x_{X_1} \in X_0$, тобто оптимальний розв'язок підзадачі допустимий у задачі (2.8). Тоді приймемо $N = W_{X_i}$ і запам'ятовуємо x_{X_i} як новий поточний розв'язок. У цьому випадку оцінка N поліпшується;

3) множина X_i не містить допустимих розв'язків.

Після перегляду всіх підмножин X_i відбираємо ті з них, які не були відкинуті. Вибираємо серед них підмножину з найбільшим значенням W_{X_i} і розбиваємо його на підмножини X_{ij}, $j = 1, \ldots, n$. Розв'язуємо підзадачі з підмножинами X_{ij}, використовуючи правила відкидання підмножин і т.д. Алгоритм завершує свою роботу, коли всі підмножини будуть відкинуті.

Зауважимо, що в кожній конкретній реалізації методу гілок і границь треба доводити його збіжність за скінченну кількість кроків. Покажемо, що при невдалому способі розгалуження задачі алгоритм може призвести до нескінченної кількості кроків. Припустимо, що в задачі цілочислового LP є єдина цілочисельна точка x^0, внутрішня для множини допустимих розв'язків X неперервної задачі. Нехай процес розбиття на підмножини проводять так, що точка x^0 завжди залишається внутрішньою для множини розбиття, що її містить. Оптимальні розв'язки підзадач, отримані симплекс-методом, належать границям підмножин розбиття. Тому в цьому прикладі метод гілок і меж не сходиться за скінченну кількість кроків.

Продемонструємо метод гілок і меж на прикладі.

Приклад 2.5 (**Задача про призначення**). Нехай $\alpha, \beta, \gamma, \delta$ – робітники, яких треба розподілити на чотири об'єкти з номерами

1, 2, 3, 4. Тут множина X_0 складається з 24 перестановок символів $\alpha, \beta, \gamma, \delta$. Наприклад, $x^1 = (\delta, \gamma, \alpha, \beta)$ означає, що робітник δ задіяний на першому об'єкті, робітник γ – на другому і т.д. У такій матриці

$$\begin{array}{c} \\ \alpha \\ \beta \\ \gamma \\ \delta \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \left(\begin{array}{cccc} 9 & 1 & 5 & 6 \\ 7 & 2 & 8 & 8 \\ 6 & 8 & 0 & 7 \\ 1 & 7 & 3 & 2 \end{array} \right) \end{array} \tag{2.9}$$

містяться величини ефективності використання робітників на кожному об'єкті. Визначимо функцію $W(x)$ як сумарну величину ефективності розподілу робочих за об'єктами. Наприклад, $W(x^1) = 1 + 8 + 5 + 8 = 22$.

Розглянемо множину X, яка складається з векторів x, за якими допускається розподіл робітників більше, ніж на один об'єкт. Множина X включає X_0 і містить 256 розв'язків.

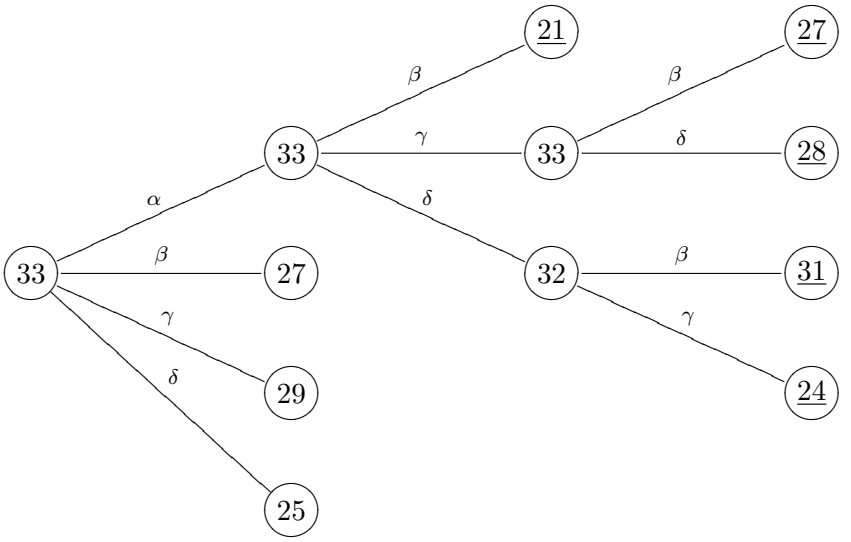
Легко бачити, що

$$\max_{x \in X} W(x) = W(x^*) = 33, \;\; x^* = (\alpha, \gamma, \beta, \beta) \notin X_0 \, .$$

Переходимо до розбиття задачі на підзадачі. Нехай X_α — множина таких x, для яких $x_1 = \alpha$, $x_j \neq \alpha$, $j = 2, 3, 4$. Інакше при $x \in X_\alpha$ лише робітник α призначається на перший об'єкт. Аналогічно визначають множини X_β, X_γ, X_δ, $X_{\alpha\beta}$ і т. п. Тоді $X = X_\alpha \cup X_\beta \cup X_\gamma \cup X_\delta \cup \tilde{X}$, де $\tilde{X}$ складається з векторів x, у яких перша компонента збігається з будь-якою іншою. Оскільки множина $\tilde{X}$ не містить допустимих розв'язків, то її можна відразу відкинути.

Алгоритм зручно проілюструвати за допомогою дерева розгалуження задачі. Вершина дерева буде відповідати деякій підмножині A множини X. Її зображають кружечком, всередині якого вказано значення W_A. Якщо $x_A \in X_0$, то значення W_A будемо підкреслювати. З кореневої вершини дерева виходить чотири ребра, відмічені символами $\alpha, \beta, \gamma, \delta$. Вони відповідають вибору підзадач з підмножинами X_α, X_β, X_γ, X_δ множини X. Наступне

за ребром α ребро β відповідає вибору підзадачі з підмножиною $X_{\alpha\beta}$ і т.д. На рисунку зображено результати обчислення. Тут $x^* = (\alpha, \delta, \beta, \gamma)$ — оптимальний розв'язок і $W(x^*) = 31$.



▲

Застосуємо метод гілок і меж для розв'язування задачі (2.1). Припустимо, що множина X допустимих розв'язків відповідної неперервної задачі (1.15) політоп. Позначимо через $[a]$ — цілу частину числа a. Нехай x^* — оптимальний розв'язок неперервної задачі (1.15). Якщо x^* містить цілі компоненти, то $x^* \in X_0$ і x^* — оптимальний розв'язок задачі (2.1). Нехай розв'язок x^* містить нецілі компоненти і x_s^* — будь-яка з них. Визначимо три множини

$$X_1 = \{x \in X : x_s \leqslant [x_s^*]\}, \quad X_2 = \{x \in X : x_s \geqslant [x_s^*] + 1\},$$

$$\tilde{X} = \{x \in X : [x_s^*] < x_s < [x_s^*] + 1\}.$$

Оскільки $\tilde{X}$ не містить допустимих розв'язків задачі (2.1), то її відразу можна відкинути. Відповідно до методу гілок і меж знаходимо оптимальні розв'язки підзадач x^1, x^2: $W_{X_1} = W(x^1)$, $W_{X_2} = W(x^2)$. Одна з множин X_1 або X_2 може виявитися порожньою, тоді її відкидаємо. Якщо $x^1 \in X_0$, то приймемо, що

$N = W_{X_1}$, розв'язок x^1 запам'ятовуємо, а множину X_1 відкидаємо. Аналогічну перевірку виконуємо для x^2. Припустимо, що обидві множини X_1 і X_2 не були відкинуті і що $W_{X_1} \geqslant W_{X_2}$. Тоді беремо множину X_1 для наступного розгалуження. Для нецілої компоненти x_k^1 визначимо дві множини $X_3 = \{x \in X_1 : x_k \leqslant [x_k^1]\}$, $X_4 = \{x \in X_1 : x_k \geqslant [x_k^1] + 1\}$. Розв'язуємо відповідні підзадачі і т. д. Алгоритм закінчує роботу за скінченну кількість кроків. Справді, при кожному розгалуженні задачі з будь-якої осі x_i відкидається інтервал (множина вигляду $\tilde{X}$). Оскільки X — політоп, то таких інтервалів скінченна кількість, і звідси випливає скінченість алгоритму.

Зауважимо, що нерівність $W_{X_1} > W_{X_2}$ не означає, що оптимальний розв'язок x^* задачі (2.1) обов'язково міститься в множині X_1.

Приклад 2.6. Розглянемо задачу

$$\begin{aligned} &\max W = \frac{1}{9}x_1 + x_2, \\ &\text{за умов} \quad x_2 \leqslant \frac{16}{9}x_1, \;\; x_1 + \frac{9}{10}x_2 \leqslant 3, \\ &\qquad\qquad x_1, x_2 \geqslant 0, \;\; x_1, x_2 \in \mathbb{Z}\,. \end{aligned}$$

Множина X_0 — допустимих розв'язків містить шість точок. Тут $x^1 = (1, 16/9)$ і $x_2 = (2, 10/9)$ — оптимальні розв'язки підзадач і $W(x^1) = 17/9 > Z(x^2) = 12/9$, але оптимальний розв'язок $x_* = (1, 2)$ задачі (2.1) належить множині X_2. ▲

Якщо множина X необмежена, то у разі застосування методу гілок і меж можливі різні перешкоди.

Приклад 2.7. Гра на фондовій біржі. Нехай A — кількість грошей, які компанія виділила для купівлі на наступних торгах n видів цінних паперів. Компанія доручає купівлю цінних паперів брокеру і задає такі обмеження. Паперів j-го вигляду можна купити на суму не більшу ніж, b_j, $j = 1, ..., n$, де

$$\sum_{j=1}^{n} b_j > A\,.$$

Крім того, є група цінних паперів J, які найбільш бажані для компанії. Вона вважає, що цінні папери з номерами $j \in J$ треба купити на суму не меншу, ніж $B > A/2$. Брокер повинен забезпечити таку купівлю цінних паперів, щоб на наступних торгах сума, виручена від їхнього продажу, була найбільшою.

Нехай a_j — вартість цінного паперу j-го вигляду в цей момент, а c_j — прогнозована брокером ціна на наступних торгах. Позначимо через x_j кількість паперів вигляду j, які купив брокер. Приймемо $u_j = [b_j/a_j]$, $j = 1, ..., n$. Для брокера виникає задача цілочислового програмування з двосторонніми обмеженнями

$$\max W = \sum_{j=1}^{n} c_j x_j,$$

$$\text{за умов} \quad \sum_{j=1}^{n} a_j x_j \leqslant A, \ \sum_{j \in J} a_j x_j \geqslant B,$$

$$x_1 \in [0, 100], \ x_2 \in [0, 160], \ x_3 \in [0, 120], \ x_j \in \mathbb{Z}, \ j = 1, 2, 3.$$

Розглянемо конкретний приклад

$$\max W = 8x_1 + 8x_2 + 5x_3,$$

$$\text{за умов} \quad 6x_1 + 8x_2 + 4x_3 \leqslant 1360, \ 6x_1 + 8x_2 \geqslant 1250,$$

$$x_1 \in [0, 100], \ x_2 \in [0, 160], \ x_3 \in [0, 120], \ x_j \in \mathbb{Z}, \ j = 1, 2, 3.$$

Розв'яжемо відповідну неперервну задачу. Введемо для першої нерівності слабку змінну x_4, а для другої — x_5.

Початкова симплекс-таблиця така:

W	-8	-8	-5	0	0	0
x_4	6	8	4	1	0	1360
x_5	-6	$\underline{-8}$	0	0	1	-1250

.

Перетворимо її, щоб отримати початковий базисний розв'язок. Для цього виконаємо операцію заміни з опорним елементом $a_{22} = -8$:

W	-2	0	-5	0	-1	1250
x_4	0	0	4	1	1	110
x_5	$3/4$	1	0	0	$-1/8$	$625/4$

.

Візьмемо опорним перший стовпець. Тоді $\theta = \min\limits_{i:\, a_{ip}>0} \dfrac{b_i}{a_{ip}} = \dfrac{625}{3} > u_1 = 100$ і, як наслідок, потрібно зробити заміну $x_1 = 100 - y_1$:

W	2	0	-5	0	-1	1450
x_4	0	0	$\underline{4}$	1	1	110
x_2	$-3/4$	1	0	0	$-1/8$	$325/4$

.

Візьмемо опорним третій стовпець. Тоді $\theta = 110/4 < u_3 = 120$ і перший рядок – опорний. Далі проводимо звичайне перетворення таблиці:

W	2	0	0	$5/4$	$1/4$	$3175/2$
x_3	0	0	1	$1/4$	$1/4$	$55/2$
x_2	$-3/4$	1	0	0	$-1/8$	$325/4$

.

Звідси знаходимо оптимальний розв'язок неперервної задачі: $y_1^* = 0$, $x_2^* = 325/4$, $x_3^* = 55/2$.

Зрештою, $x^* = (100, 325/4, 55/2)$.

Візьмемо не цілу компоненту x_2^* і розглянемо множини

$$X_1 = \{x \in X : x_2 \leqslant 81\}, \qquad X_2 = \{x \in X : x_2 \geqslant 82\},$$

де X — множина розв'язків $x = (x_1, x_2, x_3)$, які задовольняють обмеження задачі без вимоги цілочисельності компонент. Покажемо, що множина X_1 порожня. Справді, для $x \in X_1$, $6x_1 + 8x_2 \leqslant 600 + 648 = 1248 < 1250$ і друга нерівність обмеження не виконується.

Розв'яжемо підзадачу

$$\max_{x \in X_2} W(x) = W(x^*).$$

Від тільки що розв'язаної задачі вона відрізняється нерівністю $82 \leqslant x_2 \leqslant 100$. Використаємо вже наведені обчислення. Зробимо заміну $x_2 = \tilde{x}_2 + 82$. Тоді $0 \leqslant \tilde{x}_2 \leqslant 18$. Кінцева симплекс-таблиця набуде вигляду:

W	2	0	0	5/4	1/4	3175/2
x_3	0	0	1	1/4	1/4	55/2
x_4	−3/4	1	0	0	$\underline{-1/8}$	−3/4

.

Поточний базисний розв'язок не допустимий. Використаємо двоїстий симплекс-метод. Другий рядок — опорний, $\theta' = \min[8/3, 2] = 2$ і п'ятий стовпець — опорний. Після перетворення отримаємо симплекс-таблицю

W	1/2	2	0	5/4	0	1586
x_3	−3/2	2	1	1/4	0	26
x_4	6	−8	0	0	1	6

.

Звідси $y_1^0 = 0$, $\tilde{x}_2^0 = 0$, $x_3^0 = 26$. Зрештою, $x^* = (100, 82, 26)$. Оскільки розв'язок x^* складається з цілих компонент, то він оптимальний розв'язок вихідної задачі. ▲

Задачі

2.3. Методом гілок і меж розв'язати задачу про призначення на мінімум з матрицею затрат (2.9). Відповідь. $x^* = (\delta, \beta, \gamma, \alpha)$, $W(x^*) = 9$.

2.4. Розв'язати методом гілок і меж таку задачу цілочислового LP:

$$\begin{aligned} &\max W = 4x_1 + 5x_2 + x_3, \\ &\text{за умов} \quad 3x_1 + 2x_2 \leqslant 10,\ x_1 + 4x_2 \leqslant 11,\ 3x_1 + 3x_2 + x_3 \leqslant 13, \\ &\qquad\qquad x_1, x_2, x_3 \geqslant 0,\ x_j \in \mathbb{Z},\ j = 1, 2, 3. \end{aligned}$$

Відповідь. $x^* = (2, 2, 1)$.

2.5. Методом гілок і меж визначити несумісність обмеженої задачі

$$\begin{aligned} &\max W = -3x_1 - 2x_2, \\ &\text{за умов} \quad 1 \leqslant 3x_1 + 3x_2 \leqslant 2,\ x_1, x_2 \geqslant 0,\ x_j \in \mathbb{Z},\ j = 1, 2. \end{aligned}$$

2.6. Навести приклад задачі цілочислового LP, в якій $X_0 = \{(0,0)\}$, а цільова функція $W(x)$ на множині X не обмежена зверху.

2.7. Навести приклад задачі (2.1) з необмеженою множиною X і порожньою множиною X_0, в якій метод гілок і меж не збігається за скінченну кількість кроків.

2.8. Нехай у задачі (2.1) $X_0 \neq \emptyset$, коефіцієнти a_{ij} і b_j — раціональні числа, а цільова функція $W(x)$ не обмежена зверху на X. Довести, що функція $W(x)$ не обмежена зверху на X_0.

2.4 Задача булевого програмування

Розглянемо задачу *булевого лінійного програмування*

$$\begin{aligned} &\max W = \sum_{j=1}^{n} c_j x_j, \\ &\text{за умов} \quad \sum_{j=1}^{n} a_{ij} x_j \leqslant b_i,\ i = 1, ..., m, \\ &\qquad\qquad x_j \in \{0,1\},\ \ j = 1, ..., n. \end{aligned} \tag{2.10}$$

Зауважимо, що кожну задачу цілочислового LP з обмеженою множиною X відповідних неперервних розв'язків можна звести до задачі булевого LP. Для цього достатньо використати двійкове зображення цілих чисел

$$x_j = \sum_{r=0}^{l} y_{jr} 2^r,$$

де y_{jr} — булеві змінні. Оскільки x_j набуває своїх значень з деякого відрізка, то номер l можна вважати фіксованим. Однак отримана задача булевого LP може мати велику розмірність.

Наведемо приклади задачі булевого LP.

Приклад 2.8. Задача про рюкзак

Нехай фірма має капітал I, призначений для інвестування в n проектів. У цьому разі j-й проект дає прибуток V_j і потребує інвестицій у розмірі I_j. Потрібно так вибрати проекти, щоб оптимізувати сумарний прибуток.

Уведемо булеві змінні x_j, $j = 1, ..., n$. Значення $x_j = 1$ ($x_j = 0$) засвідчує, що фірма (не) інвестує капітал в j-й проект. Отримуємо задачу про рюкзак

$$\begin{aligned} &\max \sum_{j=1}^{n} V_j x_j, \\ \text{за умов} \quad &\sum_{j=1}^{n} I_j x_j \leqslant I, \\ &x_j \in \{0,1\}, \quad j = 1, ..., n. \end{aligned} \tag{2.11}$$

Спочатку вона мала таку інтерпретацію: турист збирається в похід і відбирає в рюкзак найбільш цінні для себе предмети, сумарний обсяг яких обмежений місткістю рюкзака. ▲

Приклад 2.9. Задача водопровідника. Водопровідник отримав наряд на встановлення кранів на декількох трубах, прокладених під землею, покритою плитами (рис. 2.4.1). Він може встановити механізми перекриття в будь-якому місці труби, але, звичайно, по одному крану на кожній трубі. Для мінімізації праці водопровідникові треба визначити мінімальну кількість плит, які треба підняти, щоб встановити по одному крану на кожній трубі. Пронумеруємо плити від 1 до 12, а труби від (1) до (5). Нехай булева змінна x_j набуває значення 1 або 0 залежно від того, підняли j-у плиту чи ні. Кожній трубі відповідає одне обмеження, яке означає, що для отримання до неї доступу необхідно підняти принаймні одну з покриваючих її плит. Для труби (1) будемо мати обмеження $x_1 + x_2 + x_3 \geqslant 1$. Для труб (2) – (5) можна записати аналогічні обмеження. У підсумку отримуємо задачу булевого LP

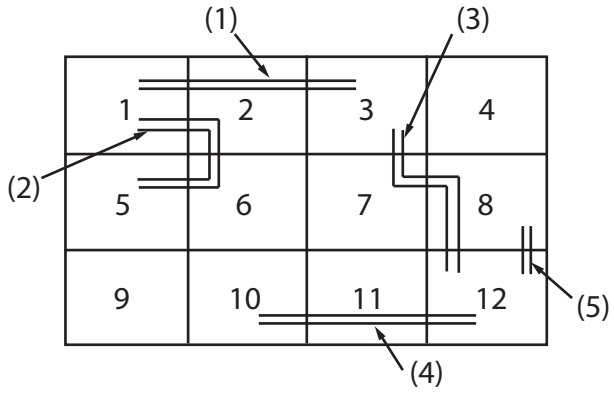


Рис. 2.4.1.

$$\begin{aligned} &\min W = \sum_{j=1}^{12} x_j, \\ &\text{за умов} \quad x_1 + x_2 + x_3 \geqslant 1, \qquad x_1 + x_2 + x_5 + x_6 \geqslant 1, \\ &\qquad\quad x_3 + x_7 + x_8 + x_{12} \geqslant 1, \quad x_{10} + x_{11} + x_{12} \geqslant 1, \\ &\qquad\quad x_8 + x_{12} \geqslant 1, \end{aligned} \tag{2.12}$$

де x_j, $j = 1, ..., 12$, — булеві змінні. ▲

Припустимо, необмежуючи загальності, що $0 \leqslant c_1 \leqslant c_2 \leqslant ... \leqslant c_n$. Справді, якщо $c_j < 0$, то після заміни змінної $x_j = 1 - y_j$ отримаємо додатний коефіцієнт $-c_j$ при y_j.

Для задачі булевого LP розглянемо **алгоритм Балаша**, який є варіантом методу гілок і меж. Нехай X_0 – множина допустимих розв'язків задачі (2.10), а X – множина всіх 2^n булевих розв'язків x. На початку роботи алгоритму перевіримо на допустимість два розв'язки: $x^* = (1, ..., 1)$ і $\overline{x} = (0, 1, ..., 1)$. Легко бачити, що x^* — оптимальний розв'язок задачі максимізації функції $W(x)$ на множині X, а $\overline{x}$ — розв'язок, на якому цільова функція набуває

друге за величиною значення. Якщо хоча б одне з них допустиме, то отримуємо розв'язок задачі (2.10). Припустимо, що обидва розв'язки не є допустимими в задачі (2.10).

Візьмемо натуральне число $k < n$. У наборах значень булевих змінних вигляду $x_1, x_2, ..., x_k$ коми часто будемо опускати. Для довільного такого набору $x_1x_2...x_k$ визначимо множину розв'язків

$$X(x_1...x_k) = \{y \in X : y_j = x_j, \quad j = 1, ..., k\}.$$

Розгалуження задачі проводиться звичайним методом: множина X розбивається на дві підмножини $X(0)$ і $X(1)$, кожна з яких розбивається відповідно на дві підмножини: $X(0) = X(00) \cup X(01)$ і $X(1) = X(10) \cup X(11)$ і т.д.

Розглянемо детально дії алгоритму, проведені для невідкинутої множини $X(x_1...x_k)$. Перед усім перевіряють допустимість розв'язку $x' = (x_1, ..., x_k, 0, 1, 1, ..., 1)$. Якщо він допустимий, то при $W(x') > N$ приймемо $N = W(x')$, розв'язок x' запам'ятовують як поточний, а множину $X(x_1, ..., x_k)$ відкидають. Зауважимо, що розв'язок $x'' = (x_1, ..., x_k, 1, 1, ..., 1)$ не є допустимим, коли огляд множин відбувається таким методом. Справді, якщо б розв'язок x'' був допустимим, то це було би визначено на якомусь попередньому кроці алгоритму при огляді множини $X(x_1, ..., x_l)$, де $x_{l+1} = 0$ — останній нуль з набору $x_1...x_k$. У цьому випадку множина $X(x_1...x_l)$ була б відкинута і, відповідно, множина $X(x_1...x_k)$ не повинна була виникнути. Коли набір $x_1...x_k$ складається лише з одиниць, то розв'язок x'' також містить лише одиниці і перевірка його на допустимість виконується на самому початку роботи алгоритму.

Отже, в множині $X(x_1...x_k)$ розв'язок x'' має бути відкинутим. Тоді максимум цільової функції на розв'язках, що залишилися, задається формулою

$$W_{X(x_1...x_k)} = W(x') = \sum_{j=1}^{k} c_j x_j + \sum_{j=k+2}^{n} c_j.$$

Це отримують з впорядкованості коефіцієнтів цільової функції.

Обговоримо додаткові дії, які дають змогу скоротити обчислення за алгоритмом. Розглянемо нерівності

$$\min_{y\in X(x_1...x_k)} \sum_{j=1}^{n} a_{ij}y_j \leqslant b_i, \quad i = 1, ..., m, \tag{2.13}$$

де коефіцієнти a_{ij}, b_i взяті з обмеження задачі булевого програмування (2.10). Легко перевірити кожну з нерівностей (2.13), оскільки мінімум у лівій частині знаходимо просто

$$\min_{y\in X(x_1...x_k)} \sum_{j=1}^{n} a_{ij}y_j = \sum_{j=1}^{k} a_{ij}x_j + \sum_{j=k+1}^{n} \min\,[a_{ij}, 0].$$

Нехай N — поточне значення оцінки максимуму задачі (2.10). Якщо множина $X(x_1, ..., x_k)$ виявилася невідкинутою, то корисно перевірити нерівність

$$N \geqslant W_{X(x_1...x_k)} - c_{k+2}. \tag{2.14}$$

Теорема 2.2. *Якщо для множини $X(x_1...x_k)$ виконується нерівність (2.14), то всередині неї достатньо обмежитися оглядом підмножин*

$$X(x_1...x_k \underbrace{1...1}_{l\ раз}), \quad l = 1, ..., n-k-1.$$

Доведення. За умови (2.14) множину $X(x_1...x_k0)$ можна не розглядати, оскільки $N \geqslant W_{X(x_1...x_k0)} - c_{k+2} = W_{X(x_1...x_k0)}$. Отже, при розгалуженні задачі можна відразу перейти до множини $X(x_1...x_k1)$. З нерівності $W_{X(x_1...x_k1)} = W_{(x_1...x_k)} + c_{k+1} - c_{k+2} \leqslant W_{X(x_1...x_k)}$ випливає, що $W_{X(x_1...x_k)} - c_{k+2} \geqslant W_{X(x_1...x_k1)} - c_{k+3}$. Це означає, що нерівність (2.14) правильна і для множини $X(x_1...x_k1)$. Міркуючи аналогічно, переконуємося, що достатньо розглянути лише множини $X(x_1...x_k \underbrace{11...1}_{l\ \text{раз}})$ і перевірити допустимість розв'язку

$$(x_1, ..., x_k, 1, ..., 1, 0, \underbrace{1, 1, ..., 1}_{l\ \text{раз}}), \quad l = 1, ..., n-k-1.$$

Для тих з них, які виявилися допустимими, необхідно перевірити, чи поліпшують вони оцінку N. Тепер множину $X(x_1...x_k)$ можна відкинути. □

У всьому іншому алгоритм Балаша є методом гілок і меж.

Приклад 2.10. У банк надійшли заявки на кредити від шести клієнтів. Нехай очікуваний прибуток у разі укладення договорів з j-м клієнтом становить c_j ум. од., де $c_1 = 3$, $c_2 = c_3 = 3,5$, $c_4 = 4$, $c_5 = c_6 = 4,5$. Розглянемо обмеження. Клієнти 1, 3 та 5-й були в боржниках, клієнти 2, 4 і 6-й — нові, невідомі, 1, 2 і 3-й подали заявки на короткострокові кредити, а 4, 5 і 6-й — на довгострокові. Керівництво банку прийняло рішення задовольнити не більше двох заявок для кожної з перерахованих груп клієнтів.

У цьому прикладі розв'язок $x = (x_1, ..., x_6)$ складається з булевих компонентів x_j, у цьому разі $x_j = 1$, якщо заявка на кредит j-му клієнтові задоволена, і $x_j = 0$ — у протилежному випадку. Отримаємо задачу булевого програмування

$$\begin{aligned} \max W = {} & 3x_1 + 3,5x_2 + 3,5x_3 + 4x_4 + 4,5x_5 + 4,5x_6, \\ \text{за умов} \quad & x_1 + x_3 + x_5 \leqslant 2, \quad x_2 + x_4 + x_6 \leqslant 2, \\ & x_1 + x_2 + x_3 \leqslant 2, \quad x_4 + x_5 + x_6 \leqslant 2, \\ & x_j \in \{0, 1\}, \quad j = 1, ..., 6. \end{aligned} \tag{2.15}$$

Застосуємо алгоритм Балаша. Обчислення зведені в табл. 2.1. У її першому стовпці розташовані номери (у порядку розгляду) множин $X(x_1, ..., x_k)$, у другому — оптимальні розв'язки відповіднинх підзадач $x_1...x_k(01...1)$, в третьому — максимуми $W_{X(x_1...x_k)}$ цільової функції для підзадач, у четвертому — список номерів невідкинутих множин $X(x_1...x_k)$. У цьому списку будемо підкреслювати номер множини, що має найбільший максимум і вибраної для розгалуження задачі. Якщо множина $X(x_1...x_k)$ містить допустимий розв'язок, то оцінку $W_{X(x_1..x_k)}$ будемо відзначати зірочкою.

При перегляді множини $X(110)$ з номером 5 отримали оцінку $N = 15,5$, яка досягається на допустимому розв'язку $(1, 1, 0, 0, 1, 1)$. Множина з номером 5 відкидається. Множина $X(111)$ з номером

Табл. 2.1.

Номер множини	$x_1...x_k(01...1)$	$W_{X(x_1...x_k)}$	Невідкинуті множини
1	0(01111)	16,5	1, $\underline{2}$
2	1(01111)	19,5	
3	10(0111)	16	1,3,$\underline{4}$
4	11(0111)	19,5	
5	110(011)	15,5*	$\underline{1}$, 3, 5, 6
6	111(011)	19	

Табл. 2.2.

j	1	2	3	4	5	6	7
I_j	1	1	2	2	2	3	4
V_j	73	89	137	157	159	211	299

6 відкидається за умови (2.13). Для множини $x(0)$ з номером 1 виконується умова (2.14): $N = 15,5 \geqslant W_{X(0)} - c_3 = 13$. Відповідно до теореми 2.2 перевіримо допустимість розв'язку $(0,1,0,1,1,1)$, $x^0 = (0,1,1,0,1,1)$, $(0,1,1,1,0,1)$ і $(0,1,1,1,1,0)$. Другий вектор x^* є допустимим і $W(x^*) = 16$. Тому $N = 16$ і множину 3 можна відкинути. Отже, $x^* = (0,1,1,0,1,1)$ — оптимальний розв'язок задачі. ▲

Задачі

2.9. Розв'язати задачу про рюкзак при $n = 7$, $I = 10$ зі значеннями решти параметрів, заданими в табл. 2.2.

Відповідь. $x^* = (1,1,0,1,1,0,1)$.

2.10. Розв'язати задачу водопровідника, використовуючи алгоритм Балаша.

Відповідь. $x^* = (1,0,0,0,0,0,0,0,0,0,0,1),\ W(x^*) = 2.$

2.11. Задача про рюкзак. Нехай є n неподільних продуктів з номерами $i = 1,\dots,n$. Вага i-го продукту становить c_i, а цінність p_i. Треба вибрати сукупність продуктів з мінімальною загальною вагою, якщо загальна цінність не менша від заданої величини A.

Математичне формулювання задачі

$$\begin{aligned} \min W(x) &= \sum_{i=1}^{n} c_i x_i, \\ \text{за умов} \quad &\sum_{i=1}^{n} p_i x_i \geqslant A, \\ & x_i \in \{0,1\},\ \ i = 1,\dots,n. \end{aligned}$$

Розв'язати задачу про рюкзак із загальною цінністю набору продуктів $A = 50$ і $c = (4,6,10,5,4,1), p = (12,15,10,16,8,5)$.

Відповідь. $x^* = (1,1,0,1,1,0)$.

Розділ 3

Динамічне програмування

3.1 Метод динамічного програмування

У задачах лінійного та нелінійного програмування керований процес, стосовно якого приймається рішення, вважається статичним, тобто незалежним від часу, тому оптимальне рішення (керування) знаходять лише на один етап планування. Такі задачі називаються *одноетапними* або *однокроковими*.

У задачах динамічного програмування керований процес залежить від часу, тому приймають рішення (для кожного етапу), що забезпечують оптимальний розвиток всього процесу в цілому. Задачі динамічного програмування називають *багатоетапними* або *багатокроковими*. Динамічне програмування —це математичний апарат, який дає підстави реалізувати оптимальне керування багатокроковими процесами прийняття рішень. Зауважимо, що досить часто шляхом деякого переформулювання певні статичні задачі можуть бути зведені до багатоетапних.

Основним методом динамічного програмування є розроблений американським математиком Р.Белманом та його учнями метод рекурентних співвідношень, відомий як метод функціональних рівнянь Белмана, в основу якого покладено такий **прин-**

цип оптимальності: *якщо керування процесом є оптимальним, то воно буде оптимальним і для процесу, що залишається після першого кроку.* У більш загальному вигляді принцип оптимальності формулюється так: *яким би не був стан системи перед черговим кроком, керування на цьому кроці потрібно вибирати так, щоб оптимізувати результат на цьому кроці плюс на всіх наступних кроках.* Цей принцип, який правильний для так званих адитивних і мультиплікативних показників керування (цільових функцій), допомагає визначити співвідношення між екстремальними значеннями цільової функції в задачах з різною тривалістю процесу та різними початковими станами.

Розглянемо цей принцип на такому прикладі: є n підприємств, k-те підприємство дає прибуток $\pi_k(x_k),\ k = 1, \ldots, n$, якщо йому виділено x_k одиниць ресурсу. Треба наявні A одиниць ресурсу розподілити між n підприємствами так, щоб сумарний прибуток був максимальним.

Нехай $x_k,\ k = 1, \ldots, n$ — кількість ресурсу, яку виділяють k-му підприємству. Тоді з формального погляду задача зводиться до задачі нелінійного програмування

$$\max W(x_1, \ldots, x_n) = \sum_{k=1}^{n} \pi_k(x_k),$$

$$\text{за умов} \qquad \sum_{k=1}^{n} x_k = A, \quad x_k \geqslant 0,\ k = 1, \ldots, n.$$

Зрозуміло, що для розв'язання цієї задачі можуть бути застосовані методи нелінійного програмування.

Розглянемо цю задачу як задачу n-етапного керування, k-им кроком якої є виділення ресурсу x_k k-му підприємству. Зауважимо, що в нашому прикладі цільова функція $W(x_1, \ldots, x_n)$ (показник ефективності керування) є сумою прибутків за всіма окремими кроками. Така функція називається *адитивною.*

Згідно з принципом оптимальності керування на кожному кроці потрібно вибирати з врахуванням його майбутніх наслідків на наступних кроках. Винятком є останній, n-й крок, після якого інших кроків немає. Його можна планувати так, щоб він сам

собою приніс найбільший прибуток. Тому процес динамічного програмування розгортається від кінця до початку: першим планується останній n-й крок. Оскільки невідомо, чим закінчився передостанній крок, то потрібно зробити припущення про закінчення $(n-1)$-го кроку і для кожного з них знайти таке керування за якого ми знайдемо умовне оптимальне керування на n-му кроці.

Тепер можна оптимізувати керування на $(n-1)$-му кроці. Зробивши усі можливі припущення про те, як закінчився $(n-2)$-й крок, для кожного з них знаходимо оптимальне рішення для $(n-1)$-го кроку, щоб прибуток за два останні кроки ($(n-1)$-й та n-й) був максимальним. Далі оптимізується рішення на $(n-2)$-му кроці і так продовжуємо, аж поки не оптимізуємо рішення на першому кроці.

Іншими словами, на кожному кроці знаходимо таке рішення, яке забезпечує оптимальне продовження процесу стосовно досягнутого на цей момент стану. У цьому разі на першому кроці стан системи відомий (A одиниць ресурсу), тому легко знаходимо оптимальне керування для першого етапу (оптимальний об'єм ресурсу x_1^*, який виділяється першому підприємству). У підсумку для наступних етапів залишається $(A - x_1^*)$ одиниць ресурсу. Використовуючи умовне оптимальне керування для другого етапу, знаходимо оптимальний об'єм ресурсу x_2^* для другого підприємства, продовжуючи так крок за кроком, знаходимо оптимальне керування процесом $x = (x_1^*, \ldots, x_n^*)$, що надає максимальне значення цільової функції $W(x_1, \ldots, x_n)$.

Отже, в процесі оптимізації керування методом динамічного програмування багатокроковий процес відбувається двічі: спочатку від кінця до початку, внаслідок чого знаходять умовні оптимальні керування на кожному кроці, другий раз — від початку до кінця, внаслідок чого знаходять оптимальні рішення на всіх кроках.

Зауважимо, що аналогічний підхід можна застосувати і у випадку мультиплікативної цільової функції, тобто, коли загальний прибуток дорівнює добутку прибутків на кожному з етапів (за умов їхньої додатності).

3.2 Рівняння Белмана

Спочатку розглянемо загальний *динамічний процес* прийняття рішень, який відбувається за n кроків. Кожному кроку k відповідає множина *станів* S_k. Якщо процес перебуває в стані $s_k \in S_k$, то приймаємо рішення про вибір *альтернативи* $x_k \in X_k(s_k)$. Після цього процес переходить у стан $s_{k+1} = \sigma_k(s_k, x_k)$, що відповідає такому $(k+1)$-му кроку, який обчислюють за функцією $\sigma_k(s_k, x_k)$. Для простоти будемо вважати, що на першому кроці множина S_1 містить єдиний стан s_1. Процес закінчується в стані $s_{n+1} \in S_{n+1}$, де вибір альтернатив не проводиться. Для кожної пари s_k, x_k, визначена оцінка $f_k(s_k, x_k)$ альтернативи x_k в стані s_k. *Найкраща альтернатива* $x_k^*(s_k)$ визначається з умови

$$f_k^*(s_k) := \max_{x_k \in X_k(s_k)} f_k(s_k, x_k) = f_k(s_k, x_k^*(s_k)).$$

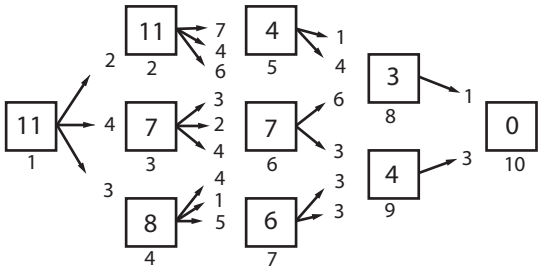
Часто замість максимуму тут пишуть мінімум, якщо це відповідає змісту оцінок.

У методі динамічного програмування передбачено, що $f_k^*(s_k)$ є найліпшим результатом, який можна забезпечити за умови, що, починаючи з k-го кроку, застосовують найкращі альтернативи. Тому $f_k(s_k, x_k)$ — оцінка альтернативи $x_k \in X_k(s_k)$ за умови, що, починаючи з $(k+1)$-го кроку, застосовують найліпші альтернативи. Звідси отримуємо, що функція $f_k(s_k, x_k)$ повинна залежати від $f_{k+1}^*(s_{k+1})$, де $s_{k+1} = \sigma_k(s_k, x_k)$. У цьому випадку $f_k^*(s_k)$ називається *функцією Белмана.*

Приклад 3.1. На рис. 3.2.1 показано розташування десяти міст, зв'язаних між собою мережею автомобільних доріг (цифри поруч зі стрілками — довжини доріг у кілометрах, під квадратиками — номери міст). Передбачено, що дорогами можна переміщатися лише в напрямах, зазначених стрілками. Треба проїхати з міста 1 в місто 10 найкоротшим шляхом.

У цьому природно вважати, що процес переміщення з міста 1 в місто 10 відбувається протягом чотирьох кроків. На першому кроці $S_1 = \{1\}$. На другому множина станів дорівнює $S_2 = \{2, 3, 4\}$ і т. д. На k-му кроці нехай x_k — дорога, яка веде

Рис. 3.2.1.



з міста s_k у наступне місто $s_{k+1} = \sigma_k(s_k, x_k)$. Уведемо функцію оцінки $f_k(s_k, x_k)$, яка визначає довжинину найкоротшого шляху з міста s_k в місто 10, якщо спочатку їхати дорогою x_k. Тоді функція Белмана

$$f_k^*(s_k) = \min_{x_k \in X_k(s_k)} f_k(s_k, x_k)$$

— довжина найкоротшого шляху з міста s_k в місто 10. Зауважимо, що

$$f_k(s_k, x_k) = d(x_k) + f_{k+1}^*(s_{k+1}),$$

де $d(x_k)$ – довжина дороги x_k. Вважаючи $f_5^*(10) = 0$, послідовно знаходимо $f_4^*(9) = 4$, $f_4^*(8) = 3$, $f_3^*(7) = 6$ і т. д. На рис. 3.2.1 значення функцій Белмана зображено всередині квадратиків. Зокрема, $f_1^*(s_1) = 11$ – довжина найкоротшого шляху $1 \to 3 \to 5 \to 10$ з міста 1 в місто 10. ▲

Розглянутий приклад засвідчує, що функція $f_k(s_k, x_k)$ повинна набувати вигляду $f_k(s_k, x_k) = g_k(x_k, f_{k+1}^*(\sigma_k(s_k, x_k)))$. Тоді з визначення функції $f_k^*(s_k)$ отримуємо

$$f_k^*(s_k) = \max_{x_k \in X_k(s_k)} g_k(x_k, f_{k+1}^*(\sigma_k(s_k, x_k))).$$

Це співвідношення називається *рівнянням Белмана*. Функцію $f_{n+1}^*(s_{n+1})$ залежно від змісту задачі приймають тотожно рівною нулю або одиниці.

Рівняння Белмана дає змогу підрахувати функції $f_k^*(s_k)$ і $x_k^*(s_k)$, починаючи з n-го і закінчуючи першим кроком. Мета підрахунків за методом динамічного програмування — знаходження величини $f_1^*(s_1)$ і послідовності $x_1^* = x_1^*(s_1)$, $s_2^* = \sigma_1(s_1, x_1^*)$, $x_2^* = x_2^*(s_2^*), ..., s_n^* = \sigma_{n-1}(s_{n-1}^*, x_{n-1}^*)$, $x_n^* = x_n^*(s_n^*)$, яка містить оптимальні значення альтернатив x_k^* для всіх кроків процесу.

Зазначимотаке: коли знайдена функція $f_k^*(s_k)$, то функцію $f_{k+1}^*(s_{k+1})$ можна "забути". Проте всі функції $x_k^*(s_k)$ треба пам'ятати, що розглядається як недолік методу. Для його застосування до будь-якої задачі необхідно конкретизувати елементи розглянутого динамічного процесу і пояснити зміст функцій Беллмана.

Розглянемо задачу максимізації адитивної цільової функції

$$\begin{aligned} &\max W = \sum_{j=1}^{n} p_j(x_j), \\ &\text{за умов} \quad \sum_{j=1}^{n} a_j x_j \leqslant A, \quad x_j \geqslant 0, \quad x_j \in \mathbb{Z}, j = 1, ..., n, \end{aligned} \tag{3.1}$$

де константи A, a_j, $j = 1, ..., n$, — додатні, а $p_j(x_j)$ — зростаючі функції цілого аргументу.

Задачу (3.1) можна інтерпретувати як **задачу оптимального розміщення рекламних оголошень**. *Нехай деяка фірма проводить рекламну компанію. Є n пунктів можливого розміщення реклами (газети, радіо, телебачення тощо). Дослідження виявили таке: якщо в j-му пункті оголошення дається t разів, то очікувана кількість клієнтів фірми збільшується на величину $p_j(t)$. Нехай A – кількість грошей, виділених на рекламу, a_j – вартість одного рекламного оголошення в j-му пункті розміщення.*

Розглянемо окремий випадок, коли $a_j = 1$, $j = 1, ..., n$, а A – ціле число. Тоді максимум у задачі (3.1) досягається, коли сума

змінних в обмеженні дорівнює A. У підсумку отримаємо задачу

$$\begin{aligned} &\max W = \sum_{j=1}^{n} p_j(x_j), \\ &\text{за умов} \quad \sum_{j=1}^{n} x_j = A, \quad x_j \geqslant 0, \quad x_j \in \mathbb{Z}, \quad j = 1, ..., n. \end{aligned} \tag{3.2}$$

Задачі (3.2) можна надати іншу інтерпретацію. *Страхова компанія розподіляє A своїх агентів за n районами міста. Якщо в j-му районі працює t агентів, то $p_j(t)$ – очікувана кількість підписаних договорів страхування. Треба так розподілити агентів за районами, щоб загальна очікувана кількість договорів було найбільшою.*

Застосуємо метод динамічного програмування до задачі (3.1). Будемо вважати, що значення змінних вибирають послідовно: спочатку x_1, потім x_2 і т.д. Приймемо

$$s_k = A - \sum_{j=1}^{k-1} a_j x_j$$

— кількість грошей, що залишилися після розміщення реклами на перших $k-1$ пунктах. У цьому разі $s_k \in S_k \subset [0, A]$, $s_{k+1} = \sigma_k(s_k, x_k) = s_k - a_k x_k$. Зазначимо, що під час виконання обмежень задачі (3.1) величина s_k набуває лише скінченну кількість значень. Зокрема, в задачі (3.2) $S_k = \{0, ..., A\}$. Далі $x_k \in X_k(s_k) = \{0, 1, ..., [s_k/a_k]\}$. Визначимо множину

$$D_k(s_k) = \left\{(x_k, ..., x_n) : \sum_{j=k}^{n} a_j x_j \leqslant s_k,\ x_j \geqslant 0,\ x_j \in \mathbb{Z},\ j = k, ..., n\right\}$$

і функцію, яка відповідає k-му кроку,

$$f_k^*(s_k) = \max_{(x_k,...,x_n) \in D_k(s_k)} \sum_{j=k}^{n} p_j(x_j). \tag{3.3}$$

Розв'язуючи задачу (3.3), фірма найкраще розподіляє гроші в кількості s_k за пунктами з номерами $k, k+1, ..., n$.

Тепер визначимо функцію $f_k(s_k, x_k) = p_k(x_k) + f^*_{k+1}(s_k - a_k x_k)$, що оцінює вибір x_k в стані s_k. Ця оцінка ґрунтується на тому, що після вибору x_k сума, яка залишилася $s_k - a_k x_k$, розподіляється оптимально за пунктами з номерами $k+1, ..., n$. Будемо вважати, що $f^*_{n+1}(s_{n+1}) \equiv 0$. Покажемо, що функції $f^*_k(s_k)$ задовольняють рівняння Белмана

$$f^*_k(s_k) = \max_{x_k \in X_k(s_k)} (p_k(x_k) + f^*_{k+1}(s_k - a_k x_k)), \quad k = n, n-1, ..., 1.$$

При $k = n$

$$f^*_n(s_n) = \max_{x_n \in X_n(s_n} p_n(x_n) = p_n([s_n/a_n]), \quad x^*_n(s_n) = [s_n/a_n]$$

і твердження очевидне.

Нехай $k < n$. Візьмемо вектор $(x^*_k, ..., x^*_n) \in D_k(s_k)$ такий, що

$$f^*_k(s_k) = \sum_{j=k}^{n} p_j(x^*_j).$$

Тоді $x^*_k \in X_k(s_k)$ і $(x^*_{k+1}, \ldots, x^*_n) \in D_{k+1}(s_k - a_k x^*_k)$. Звідси

Табл. 3.1.

t	$p_1(t)$	$p_2(t)$	$p_3(t)$
0	0	0	0
1	20	30	25
2	40	54	50
3	55	82	70
4	70	100	90

$$f^*_k(s_k) \leqslant \max_{x_k \in X_k(s_k)} (p_k(x_k) + f^*_{k+1}(s_k - a_k x_k)).$$

Аналогічно доводимо, що

$$f_k^*(s_k) \geqslant \max_{x_k \in X_k(s_k)} (p_k(x_k) + f_{k+1}^*(s_k - a_k x_k)).$$

З рівняння Белмана послідовно знаходимо функції $f_k^*(s_k)$, $x_k^*(s_k)$, $k = n-1, \ldots, 1$. Оскільки $s_1 = A$, то $s_2 = A - a_1 x_1^*(A)$, $s_3 = s_2 - a_2 x_2^*(s_2), \ldots$ Звідси отримуємо $x^* = (x_1^*(A), x_2^*(s_2), \ldots, x_n^*(s_n))$ — оптимальний розв'язок задачі (3.1).

Якщо в задачі (3.1) числа n і A великі, то у разі реалізації методу динамічного програмування збереження функцій $f_k^*(s_k)$ і $x_k^*(s_k)$ може потребувати великого об'єму пам'яті. Цих труднощів менше в задачі (3.2), де величина s_k при кожному k набуває значення $0, 1, \ldots, A$. Аналогічні спрощення можливі й у випадку, коли всі величини a_j цілі.

Приклад 3.2. Розміщення реклами. Організація вирішила розмістити рекламні оголошення в газеті, на телебаченні та по радіо. Вартість одного оголошення в газеті становить 220 у.о., на телебаченні – 300 і на радіо – 250 у.о. Витрати на рекламу заплановано в розмірі 1000 у.о. Нехай $p_j(t)$ – очікувана кількість нових клієнтів після t оголошень в j-му пункті розміщення реклами, $j = 1, 2, 3$. Функції $p_j(t)$ задані в табл. 3.1 Потрібно найефективніше використати гроші на рекламу. Отримуємо задачу (3.1) з $A = 1000$, $a_1 = 220$, $a_2 = 300$, $a_3 = 250$. Застосуємо метод динамічного програмування. Зауважимо, що функції $f_3^*(s_3) = p_3([s_3/250])$ і $x_3^*(s_3) = [s_3/250]$ кусково-сталі. Випишемо їхні значення в таблицю:

s_3	$f_3^*(s_3)$	$x_3^*(s_3)$
$0 \leqslant s_3 < 250$	0	0
$250 \leqslant s_3 < 500$	25	1
$500 \leqslant s_3 < 750$	50	2
$750 \leqslant s_3 < 1000$	70	3
$s_3 = 1000$	90	4

Тепер протабулюємо функції

$$f_2(s_2, x_2) = p_2(x_2) + f_3^*(s_2 - 300x_2),$$

$$f_2^*(s_2) = \max_{x_2 \in X_2(s_2)} f_2(s_2, x_2) \text{ і } x_2^*(s_2).$$

Зауважимо, що величина $s_2 = 1000 - 220x_1 \in S_2 = \{120, 340, 560, 780, 1000\}$.

Випишемо таблицю:

s_2	$f_2(s_2,0)$	$f_2(s_2,1)$	$f_2(s_2,2)$	$f_2(s_2,3)$	$f_2^*(s_2)$	$x_2^*(s_2)$
120	0				0	0
340	25	30			30	1
560	50	55			55	1
780	70	55	54		70	0
1000	90	80	79	82	90	0

Знайдемо

$$f_1(s_1, x_1) = p_1(x_1) + f_2^*(s_1 - 220x_1),$$

$$f_1^*(s_1) = \max_{x_1 \in X_1(s_1)} f_1(s_1, x_1),\ x_1^*(s_1)$$

при одному значенні $s_1 = 1000$. Отримаємо таблицю:

$f_1(s_1,0)$	$f_1(s_1,1)$	$f_1(s_1,2)$	$f_1(s_1,3)$	$f_1(s_1,4)$	$f_1^*(s_1)$	$x_1^*(s_1)$
90	90	95	85	70	95	2

Звідси знаходимо $x_1^* = x_1^*(s_1) = 2$; $s_2^* = s_1 - 2 \cdot 220 = 560$; $x_2^* = x_2^*(560) = 1$; $s_3^* = 560 - 1 \cdot 300 = 260$; $x_3^* = x_3^*(260) = 1$. Отже, $x^* = (2, 1, 1)$ — оптимальний розв'язок задачі. Організації треба розмістити два оголошення в газеті і по одному на телебаченні та на радіо. ▲

Приклад 3.3. Розглянемо задачу

$$\begin{aligned} &\max W = \prod_{j=1}^{n} p_j(x_j), \\ &\text{за умов} \quad \sum_{j=1}^{n} a_j(x_j) \leqslant A, \\ &\qquad 1 \leqslant x_j \leqslant m, \quad x_j \in \mathbb{Z}, \quad j = 1, ..., n, \end{aligned} \tag{3.4}$$

де $p_j(x_j), a_j(x_j)$ — зростаючі функції цілого аргументу.

Задача (3.4) має таку інтерпретацію. *Проектується система, яка складається з n елементів. Кожен j-й елемент може бути дубльований. У цьому разі число x_j паралельно з'єднаних елементів не може бути більше m; $p_j(x_j)$ — ймовірність невиходу з ладу j-го дубльованого елемента, а $a_j(x_j)$ — його вартість. Тоді W — ймовірність не виходу з ладу всієї системи (її надійність). Нехай A — сума грошових коштів, призначених для створення системи. Потрібно спроектувати систему, яка має найбільшу надійність.*

Застосуємо метод динамічного програмування. Нехай

$$s_k = A - \sum_{j=1}^{k-1} a_j(x_j)$$

— сума коштів, яка залишилася після вибору значень $x_1, ..., x_{k-1}$ на попередніх кроках. У цьому разі

$$s_k \in S_k \subset \Big[\sum_{j=k}^{n} a_j(1), A - \sum_{j=1}^{k-1} a_j(1)\Big].$$

Для $s_k \in S_k$ визначимо $x_k^0 = \max\{x_k : a_k(x_k) \leqslant s_k\}$. Тоді альтернатива $x_k \in X_k(s_k) = \{1, ..., x_k^0(s_k)\}$. Приймемо $f_{n+1}^*(s_{n+1}) \equiv 1$, $\sigma_k(s_k, x_k) = s_k - a_k(x_k)$,

$$f_k^*(s_k) = \max_{x_k \in X_k(s_k)} (p_k(x_k) f_{k+1}^*(s_k - a_k(x_k)))$$

— рівняння Белмана. Далі зауважимо, що

$$f_n^*(s_n) = \max_{x_n \in X_n(s_n)} p_n(x_n) = p_n(x_n^0(s_n)), \quad x_n^*(s_n) = x_n^0(s_n).$$

Як і для випадку адитивної функції, з рівняння Белмана послідовно знаходимо функції $f_k^*(s_k), x_k^*(s_k), k = n-1, n-2, ..., 1$. Далі $s_1 = A, s_2 = A - a_1(x_1^*(A)), s_3 = s_2 - a_2(x_2^*(s_2)), ...$ Звідки отримуємо $x^* = (x_1^*(A), x_2^*(s_2), ..., x_n^*(s_n))$ — оптимальний розв'язок задачі (3.4). ▲

3.3 Задача інвестування

Вісім умовних одиниць певного ресурсу (наприклад, мільйонів гривень) можуть бути інвестовані у розвиток трьох підприємств ($k = 1, 2, 3$). Позначимо через $\pi_k(x),\ k = 1, 2, 3$ прибуток у тих самих умовних одиницях, що одиниць ресурсу. Величини прибутків $\pi_k(x),\ k = 1, 2, 3$ для різних можливих значень наведені у табл. 3.2. Зауважимо, що ця таблиця містить інші дані, зміст яких надамо згодом.

Табл. 3.2.

x	0	1	2	3	4	5	6	7	8
$\pi_1(x)$	0	5	15	40	80	90	95	98	100
$\pi_2(x)$	0	5	15	40	60	70	73	74	75
$\pi_3(x)$	0	4	26	40	45	50	51	52	53
$f_3(x)$	0	4	26	40	45	50	51	52	53
$d_3(x)$	0	1	2	3	4	5	6	7	8
$f_2(x)$	0	5	26	40	60	70	86	100	110
$d_2(x)$	0	1	0	0	4	5	4	4	5

Ми вважаємо, що ресурс вимірюється лише в цілих числах і прибутки від підприємств є незалежними при будь-якому розподілі ресурсу. Треба так розділити наявні ресурси між підприємствами, щоб загальний прибуток від них був максимальним.

Нехай $x_k,\ k = 1, 2, 3$ — обсяг інвестицій у k-тее підприємство. Формулювання задачі: знайти вектор $x = (x_1, \ldots, x_3)$ такий, що

$$\max W(x_1, \ldots, x_3) = \sum_{k=1}^{3} \pi_k(x_k),$$

$$\text{за умов} \quad \sum_{k=1}^{3} x_k = 8,$$

$$x_k \geqslant 0,\ k = 1, \ldots, 3.$$

Оскільки цільова функція адитивна, то застосуємо метод динамічного програмування до розв'язування цієї задачі. Отже, розглядаємо трьохетапний процес планування, кожен з трьох етапів якого це виділення ресурсу відповідному підприємству. Зауважимо, що оптимальний розв'язок задачі не залежить від того, в якому порядку пронумерувати підприємства. Розглядаємо останній (3-й) етап планування і нехай на перших двох етапах не використано жодної одиниці ресурсу, тобто до третього етапу ми прийшли, маючи 8 одиниць ресурсу. Тоді для отримання максимального прибутку потрібно всі 8 одиниць інвестувати в розвиток третього підприємства, тому що $\pi_3(x)$ — зростаюча функція (див. табл. 3.2). У цьому разі максимальний прибуток від інвестування $f_3(8) = \pi_3(8) = 53$ одиниць і для цього треба вкласти в розвиток третього підприємства $d_3(8) = 8$ одиниць ресурсу. Для розв'язання задачі методом динамічного програмування потрібні також значення $f_3(x), d_3(x)$ при $x = 0, 1, 2, \ldots, 7$. Їх знаходять аналогічно і подано в табл.3.2. Отже, ми зробили всі можливі припущення стосовно закінчення передостаннього (2-го) етапу планування і отримали відповідні умовні оптимальні рішення для третього кроку.

Нехай тепер 8 одиниць ресурсу розподіляється між другим і третім підприємствами і z — об'єм інвестування в друге підприємство. Використовуючи значення $f_3(x)$ з табл. 3.2, легко підраховуємо

$$
\begin{aligned}
f_2(8) &= \max_{z=0,1,\ldots,8}[\pi_2(z) + f_3(8-z)] \\
&= \max[\pi_2(0) + f_3(8), \pi_2(1) + f_3(7), \pi_2(2) + f_3(6), \pi_2(3) + f_3(5), \\
&\pi_2(4) + f_3(4), \pi_2(5) + f_3(3), \pi_2(6) + f_3(2), \pi_2(7) + f_3(1), \pi_2(8) + f_3(0) \\
&= \max[53; 5+52; 15+51; 40+50; 60+45; 70+40; 73+26; 74+4; 75] \\
&= \max[53; 57; 66; 90; 105; 99; 78; 75] = 110,
\end{aligned}
$$

тобто за цих умов у розвиток другого підприємства треба вкласти $d_2(8) = 5$ одиниць ресурсу. Аналогічно знаходимо $f_2(x)$ та $d_2(x)$ при $x = 0, 1, 2, \ldots, 7$ (табл.3.2). Отже, ми отримали умовні величини прибутків для другого та третього етапів ($f_2(x)$) та

умовний оптимальний об'єм інвестування в друге підприємство ($d_2(x)$) за умов різних закінчень першого етапу.

Розгляд першого етапу планування еквівалентний розв'язанню вихідної задачі. Нехай з 8 одиниць ресурсу z одиниць інвестується у перше підприємство, $(8-z)$ - у друге та третє. Використовуючи значення $f_2(x)$ з табл. 1, знаходимо

$$f_1(8) = \max_{z=0,1,\ldots,8} [\pi_1(z) + f_2(8-z)] = \pi_1(4) + f_2(4) = 140,$$

тобто $d_1(8) = 4$. Отже, максимальний прибуток в 140 одиниць отримують при такому розподілі інвестицій: $d_1(8) = 4$ одиниці в перше підприємство, $d_2(4) = 4$ в друге підприємство та $d_3(0) = 0$ одиниць в третє.

Загальний вигляд розглянутих рівнянь такий:

$$\begin{aligned} f_3(x) &= \pi_3(x),\ d_3(x) = x, \\ f_i(x) &= \max\nolimits_{z=0,1,\ldots,8} [\pi_i(z) + f_{i+1}(x-z)], \\ d_i(x) &= \arg \max_{z=0,1,\ldots,8} [\pi_i(z) + f_{i+1}(x-z)],\ i = 2,1;\ x = 0,1,\ldots,8, \end{aligned}$$

де $f_i(x)$ — максимальний прибуток від інвестування x одиниць ресурсу в підприємства $i, i+1, \ldots, 3$, $d_i(x)$ — оптимальне інвестування в i-те підприємство, якщо в підприємства $i, i+1, \ldots, 3$ інвестується x одиниць ресурсу.

Узагальнимо наведені вище міркування. Нехай A одиниць ресурсу треба розподілити між n $(k = 1, \ldots, n)$ підприємствами, $\pi_k(x_k)$ — прибуток від k-го підприємства при інвестуванні в його розвиток x_k одиниць ресурсу, $k = 1, \ldots, n$, $W(x_1, \ldots, x_n)$ — загальний прибуток. Вважаємо, що $\pi_k(x_k),\ k = 1, \ldots, n$ — неспадні функції своїх аргументів. З математичного погляду задача зводиться до знаходження вектора $x = (x_1, \ldots, x_n)$, що є розв'язком задачі математичного програмування

$$\max W(x_1, \ldots, x_n) = \sum_{k=1}^{n} \pi_k(x_k),$$

$$\text{за умов} \quad \sum_{k=1}^{n} x_i = A,\ x_k \geqslant 0,\ k = 1, \ldots, n.$$

Приймемо

$$f_i(x) = \max_{\substack{\{x_i,\ldots,x_n\} \\ \sum_{k=i}^{n} x_k = x}} \sum_{k=1}^{n} \pi_k(x_k),\ i = n-1,\ldots,1.$$

Тоді

$$\begin{aligned} f_i(x) &= \max_{0\leqslant x_i\leqslant x}\Big[\max_{\substack{\{x_{i+1},\ldots,x_n\} \\ \sum_{k=i+1}^{n} x_k = x - x_i}} \sum_{k=1}^{n} \pi_k(x_k)\Big] \\ &= \max_{0\leqslant x_i\leqslant x}\Big[\pi_i(x_i) + \max_{\substack{\{x_{i+1},\ldots,x_n\} \\ \sum_{k=i+1}^{n} x_k = x - x_i}} \sum_{k=1}^{n} \pi_k(x_k)\Big] \\ &= \max_{0\leqslant x_i\leqslant x} \{\pi_i(x_i) + f_{i+1}(x - x_i)\},\ i = n-1,\ldots,1, \end{aligned}$$

де $x = 0, 1, \ldots, A$ для кожного $i = n-1, \ldots, 1$. Остання рівність — головний результат рекурентних співвідношень, зв'язує $f_i(x)$ та $f_{i+1}(x)$, і називається *функціональним рівнянням Беллмана.*

Нехай

$$\begin{aligned} d_i(x) &= \arg\max_{0\leqslant x_i\leqslant x} \{\pi_i(x_i) + f_{i+1}(x - x_i)\},\ i = n-1,\ldots,1, \\ d_n(x) &= \arg\max_{0\leqslant x_n\leqslant x} \pi_n(x_n) \end{aligned}$$

Тоді $f_1(A)$ — максимальний прибуток, який можна отримати від інвестування A одиниць ресурсу в розвиток n підприємств, $y_k = d_k\left(A - \sum_{l=1}^{k-1} y_l\right)$, $k = 1, \ldots, n$ — оптимальний обяг інвестицій в k-те підприємство.

Задачі

3.1. Нехай система складається з чотирьох елементів, кожен з яких може бути паралельно з'єднаний ще з двома ($m = 3$).

Для створення системи виділена сума в 100 у.о. Функції $p_j(t), a_j(t)$ задані в табл. 3.3. Знайти варіант найбільш надійної системи.

Відповідь. $x^* = (1, 2, 1, 3)$.

Табл. 3.3.

t	$p_1(t)$	$p_2(t)$	$p_3(t)$	$p_4(t)$	$a_1(t)$	$a_2(t)$	$a_3(t)$	$a_4(t)$
1	0,7	0,5	0,7	0,6	10	20	10	20
2	0,8	0,7	0,9	0,7	20	40	30	30
3	0,9	0,8	0,95	0,9	30	50	40	40

3.2. Методом динамічного програмування розв'язати задачу про оптимальний розподіл капіталовкладень з прикладу 2.10 при довільних значеннях $I \in \{1, ..., 10\}$.

3.4 Задача про заміну обладнання

На підприємстві експлуатується деякий механізм 1999 р. випуску. Розглядається задача про доцільність його заміни протягом 2001-2005 років. На початку кожного року приймається одне з двох можливих рішень: або замінити механізм (з), або продовжити його експлуатацію (е). Звичайно, експлуатація механізму приносить підприємству певний прибуток, а також потребує витрат на його утримання. Вважається також, що відома вартість заміни механізму на початку року. Перша частина таблиці містить значення цих величин для механізму 1999 р. випуску в наступні роки, розпочинаючи з 2001 р. Наприклад, у 2004 р. вік механізму становитиме 5 років, його експлуатація протягом цього року принесе прибуток у 6 одиниць, витрати на утримання становитимуть 4 одиниці, а його заміна на початку року обійдеться підприємству в 28 аналогічних одиниць.

Може трапитися, що буде доцільно замінити механізм 1999 р. на початку 2001 р. Тоді для подальшого аналізу потрібні ана-

Табл. 3.4.

	2001	2002	2003	2004	2005
Механізм 1999 р. випуску					
Вік	2	3	4	5	6
Прибуток	10	8	8	6	4
Утримання	3	3	4	4	5
Заміна	25	26	27	28	29
Механізм 2001 р. випуску					
Вік	0	1	2	3	4
Прибуток	14	16	16	14	12
Утримання	1	1	2	2	3
Заміна	20	22	24	25	26
Механізм 2002 р. випуску					
Вік		0	1	2	3
Прибуток		16	14	14	12
Утримання		1	1	2	2
Заміна		20	22	24	25
Механізм 2003 р. випуску					
Вік			0	1	2
Прибуток			18	16	16
Утримання			1	1	2
Заміна			20	22	24
Механізм 2004 р. випуску					
Вік				0	1
Прибуток				18	16
Утримання				1	1
Заміна				21	22
Механізм 2005 р. випуску					
Вік					0
Прибуток					20
Утримання					1
Заміна					21

логічні дані про механізм 2001 р. випуску. Їх подаємо в другій частині табл. 3.4.

Якщо ж заміна механізму 1999 р. відбудеться на початку 2002 р., то надалі аналізуватимуться прибутки та витрати, пов'язані з експлуатацією механізму 2002 р. випуску (третя частина таблиці 3.4) тощо. Отже, табл. 3.4, отже містить всю потрібну інформацію, пов'язану з можливою заміною механізму протягом 2001-2005 років. Зауважимо, що наведеної інформації достатньо і для щорічної заміни механізму: на початку 2001 р. замінити механізм 1999р. випуску, експлуатувати цей механізм протягом року і замінити на початку 2002 року і т. д.

Треба визначити рішення на початку кожного року так, щоб максимізувати сумарні прибутки за 2001-2005 роки. Будемо вважати 2001р.— першим роком, ..., 2005 р. — п'ятим роком планового періоду і нехай:

$r_i(t)$ — прибуток від експлуатації t-річного механізму протягом i-го року;

$u_i(t)$ — витрати на утримання протягом i-го року t-річного механізму;

$c_i(t)$ — вартість заміни t-річного механізму на початку i-го року;

$IT = 2$ — вік механізму на початку розглядуваного періоду;

$f_i(t)$ — максимальний прибуток для періоду з років $i, i + 1, \ldots, 5$ за умови, що на початку i-го року маємо t-річний механізм;

$x_i(t)$ — рішення, яке потрібно прийняти на початку i-го року для отримання $f_i(t)$.

Нехай на початок i-го року планування є t-річний механізм. У нас є дві альтернативи: замінити його або продовжувати експлуатацію. У першому випадку витрати протягом i-го року становитимуть $u_i(0)$ одиниць на утримання нового механізму та $c_i(t)$ на заміну старого механізму. У цьому разі експлуатація нового механізму протягом i-го року дає прибуток в $r_i(0)$ одиниць і на початку $i+1$-го року планування отримаємо однорічний механізм. Це означає, що у випадку заміни механізму загальний прибуток становить $r_i(0) - u_i(0) - c_i(0) + f_{i+1}(1)$ одиниць. Оскільки ми

прагнемо максимізувати прибуток, то функціональне рівняння для i-го періоду набуде вигляду

$$f_i(t) = \max\{r_i(0) - u_i(0) - c_i(0) + f_{i+1}(1); r_i(t) - u_i(t) + f_{i+1}(t+1)\},$$
$$i = 1, \ldots, 5,\ t = 1, \ldots, i-1, i+IT-1.$$

Для використання функціонального рівняння при $i = 5$ необхідні значення $f_6(j)$ для $j = 1, \ldots, 5, 7$. Приймемо їх такими, що дорівнюють нулю. Це означає, що ми не приймаємо до уваги прибутки, які утримують поза плановим періодом. Використовуючи функціональне рівняння та дані з табл. 3.4, отримаємо, наприклад,

$$\begin{aligned}
\text{для } i=5,\ t=1,\ \ f_5(1) &= \max\{r_5(0) - u_5(0) - c_5(1) + f_6(1);\\
&\quad\ r_5(1) - u_5(1) + f_6(2)\}\\
&= \max\{20 - 1 - 22 + 0;\ 16 - 1 + 0\}\\
&= \max\{-3;\ 15\} = 15 \Rightarrow x_5(1) = (e),\\
\text{для } i=5,\ t=2,\ \ f_5(2) &= \max\{r_5() - u_5(0) - c_5(2) + f_6(1);\\
&\quad\ r_5(2) - u_5(2) + f_6(3)\}\\
&= \max\{20 - 1 - 24 + 0;\ 16 - 2 + 0\}\\
&= \max\{-5;\ 14\} = 14 \Rightarrow x_5(2) = (e),\\
\text{для } i=5,\ t=3,\ \ f_5(3) &= \max\{r_5(0) - u_5(0) - c_5(3) + f_6(1);\\
&\quad\ r_5(3) - u_5(3) + f_6(4)\}\\
&= \max\{20 - 1 - 25 + 0;\ 12 - 2 + 0\}\\
&= \max\{-6;\ 10\} = 10 \Rightarrow x_5(3) = (e),\\
\text{для } i=5,\ t=4,\ \ f_5(4) &= \max\{r_5(0) - u_5(0) - c_5(4) + f_6(1);\\
&\quad\ r_5(4) - u_5(4) + f_6(5)\}\\
&= \max\{20 - 1 - 26 + 0;\ 12 - 3 + 0\}\\
&= \max\{-7;\ 9\} = 9 \Rightarrow x_5(4) = (e),\\
\text{для } i=5,\ t=6,\ \ f_5(6) &= \max\{r_5(0) - u_5(0) - c_5(6) + f_6(1);\\
&\quad\ r_5(6) - u_5(6) + f_6(7)\}\\
&= \max\{20 - 1 - 29 + 0;\ 4 - 5 + 0\}\\
&= \max\{-10;\ -1\} = -1 \Rightarrow x_5(6) = (e).
\end{aligned}$$

Табл. 3.5.

t	$f_5(t)$	$x_5(t)$	$f_4(t)$	$x_4(t)$	$f_3(t)$	$x_3(t)$	$f_2(t)$	$x_2(t)$	$f_1(t)$	$x_1(t)$
1	15	(е)	29	(е)	35	(е)	50	(е)		
2	14	(е)	22	(е)	35	(е)			38	(з)
3	10	(е)	21	(е)			24	(з,е)		
4	9	(е)			19	(з)				
5			4	(з)						
6	-1	(е)								

Одержані результати записуємо до таблиці 3.5. Виконуючи аналогічні обчислення послідовно для $i = 4,\ t = 1, 2, 3, 5$; $i = 3,\ t = 1, 2, 4, 5$; $i = 2,\ t = 1, 3, 4, 5$; $i = 1,\ t = IT$, отримуємо решту результатів табл. 3.5.

Отже, оптимальне рішення таке: замінити механізм на початку 2001 р. і експлуатувати його протягом 5 років. У цьому разі прибуток становить 38 одиниць.

3.5 Задача планування виробництва та запасів

Нехай заводу-постачальнику треба запланувати випуск продукції на наступний рік так, щоб забезпечити певний графік попиту на свою продукцію кожного місяця. Наприклад, у січні місяці потрібно поставити 150 одиниць продукції, у лютому - 70, у березні - 250 тощо.

Можна виробляти кожного місяця стільки продукції, скільки її потрібно за графіком. Однак такий план випуску пов'язаний з надмірними витратами на розширення виробництва в період підвищеного попиту та витратами на простій обладнання в період зниженого попиту з тим, щоб зберегти його та використати в період підвищеного попиту. Проте при такому плануванні виробництва варто враховувати витрати на зберігання продукції. За-

дача полягає в тому, щоб визначити план випуску продукції, при якому мінімізуються сумарні витрати, пов'язані з виробництвом та зберіганням продукції. Часто цю задачу скорочено називають задачею згладжування виробництва.

Нехай для простоти кількість періодів поставки продукції $n = 6$ і об'єми попиту такі:

Період i	1	2	3	4	5	6
Попит d_i	8	4	6	2	10	4

Будемо вважати також, що запаси продукції на початку 1-го періоду $IO = 0$, запас продукції в кінці n-го періоду $EI = 0$, вартість виробництва j одиниць продукції протягом i-го періоду позначається $PC_i(j)$ і визначається співвідношенням

$$PC_i(j) = \begin{cases} 0, & j = 0,\ i = 1, \ldots, n, \\ 20 + 5j, & j \neq 0,\ i = 1, \ldots, n, \end{cases}$$

в якому є вартість наладки виробництва, 5 є вартість виробництва одиниці продукції, вартість зберігання j одиниць продукції на кінець i-го періоду $EIC_i(j)$ визначається співвідношенням

$$EIC_i(j) = j,\ i = 1, \ldots, n,$$

тобто, витрати на зберігання визначаються кількістю продукції на кінець періоду.

Дотримуючись ідеї методу динамічного програмування, розглянемо оптимальне планування на n-й (6-й) період. Нехай $f_n(k)$, $x_n(k)$ — мінімальна вартість виробництва і його об'єм для n-го періоду за умови, що на початку n-го періоду є запас об'єму k одиниць продукції. Тоді

$$f_n(k) = \begin{cases} 0, & \text{для } k \geqslant d_n, \\ 20 + 5(d_n - k), & \text{для } k = 0, 1, \ldots, d_n - 1, \end{cases}$$

$$x_n(k) = \begin{cases} 0, & \text{для } k \geqslant d_n, \\ d_n - k, & \text{для } k = 0, 1, \ldots, d_n - 1, \end{cases}$$

Значення $f_n(k)$ і $x_n(k)$ для різних можливих значень k наведені в табл. 3.6.

Переходимо до 5-го етапу, розглядаючи оптимальне планування для пятого та шостого періодів. Якщо запас на початок пятого періоду менший, ніж d_5, то мінімальна кількість продукції, що має бути вироблена, дорівнює $d_5 - k$, інакше — 0. Аналогічно, максимальний об'єм виробництва на 5-му етапі визначається величиною $d_5 + d_6 - k$. Тому

$$\begin{aligned} f_5(k) &= \min_{\max(0,d_5-k)\leqslant z\leqslant d_5+d_6-k} [PC_5(z)+EIC_5(k+z-d_5)+f_6(k+z-d_5)] \\ &= \min_{\max(0,10-k)\leqslant z\leqslant 14-k} [PC_5(z)+k+z-10+f_6(k+z-10)]\,. \end{aligned}$$

Нехай $x_5(k)$ — розв'язок цієї задачі. Тоді для можливих значень $k = 0, 1, \ldots, d_5 + d_6 = 14$ отримуємо $f_5(k)$ та $x_5(k)$ (див. табл. 3.6 і 3.7).

Для решти етапів $i = 4, 3, 2, 1$ величини $f_i(k)$, $x_i(k)$ визначають аналогічно. Взагалі

$$f_i(k) = \min_{\max(0,d_i-k)\leqslant z\leqslant \sum\limits_{j=i}^{6} d_i-k} [PC_i(z)+EIC_i(k+z-d_i)+f_{i+1}(k+z-d_i)],$$

а $x_i(k)$ — оптимальний розв'язок цієї задачі. Величини $f_i(k)$, $x_i(k)$, $i = 1, \ldots, 6$, подані у табл. 3.6. Як бачимо, $x_1 = 20$,

$$\begin{aligned} x_2(20-d_1) &= x_2(20-8)x_2(12) = 0, \\ x_3(12-d_2) &= x_3(12-4) = x_3(8) = 0, \\ x_4(8-d_3) &= x_4(8-6) = x_4(2) = 0, \\ x_5(2-d_4) &= x_5(2-2) = x_5(0) = 14, \\ x_6(14-d_5) &= x_6(14-10) = x_6(4) = 0, \end{aligned}$$

тобто, оптимальний план виробництва у розглянутій задачі такий: протягом першого періоду виробити 20 одиниць продукції для задоволення попиту протягом перших чотирьох періодів, протягом пятого періоду виробити 14 одиниць продукції для задоволення попиту останніх двох періодів. У цьому разі мінімальні витрати становлять 236 одиниць.

Табл. 3.6.

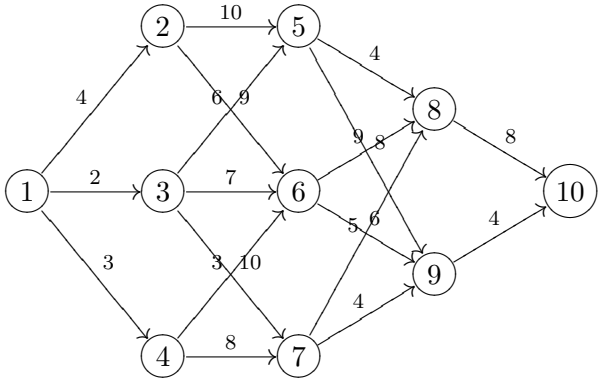
k	$f_6(k)$	$x_6(k)$	$f_5(k)$	$x_5(k)$	$f_4(k)$	$x_4(k)$	$f_3(k)$	$x_3(k)$	$f_2(k)$
0	40	4	94	14	118	16	156	8	184
1	35	3	89	13	113	15	151	7	179
2	30	2	84	12	94	0	146	6	174
3	25	1	79	11	90	0	141	5	169
4	0	0	74	10	86	0	136	4	156
5	0	0	69	9	82	0	131	3	152
6	0	0	64	8	78	0	118	0	148
7	0	0	59	7	74	0	114	0	144
8	0	0	54	6	70	0	96	0	140
9	0	0	49	5	66	0	93	0	136
10	0	0	40	0	62	0	90	0	124
11	0	0	36	0	58	0	87	0	121
12	0	0	32	0	50	0	84	0	104
13	0	0	28	0	47	0	81	0	102
14	0	0	4	0	44	0	78	0	100
15	0	0	0	0	41	0	75	0	98
16	0	0	0	0	18	0	72	0	96
17	0	0	0	0	0	0	69	0	92
18	0	0	0	0	0	0	62	0	92
19	0	0	0	0	0	0	60	0	90
20	0	0	0	0	0	0	58	0	88
21	0	0	0	0	0	0	56	0	86
22	0	0	0	0	0	0	34	0	80
23	0	0	0	0	0	0	0	0	79
24	0	0	0	0	0	0	0	0	78
25	0	0	0	0	0	0	0	0	77
26	0	0	0	0	0	0	0	0	56
27	0	0	0	0	0	0	0	0	0

Табл. 3.7.

k	$x_2(k)$	$f_1(k)$	$x_1(k)$
0	12	236	20
1	11		
2	10		
3	9		
4 - 27	0		

3.6 Задача про найкоротший шлях на мережі

Треба знайти найкоротший шлях з вершини 1 до вершини 10 на графі, зображеному на рис.3.5. Цифри на дугах означають довжини c_{ij} відповідних комунікацій.



(3.5)

Етап 1 Етап 2 Етап 3 Етап 4 Етап 5

Застосуємо ідею методу динамічного програмування до цієї задачі. Етапи позначені на рис.3.5. Нехай $f_k(x)$ - мінімальна відстань від вершини x, що належить k-му етапу, до вершини 10, $d_k(x)$ — вершина, до якої треба перейти після вершини x, дотримуючись шляху мінімальної довжини. Якщо ми перебуваємо на

Табл. 3.8.

x	1	2	3	4	5	6	7	8	9
$f_4(x)$								8	4
$d_4(x)$								10	10
$f_3(x)$					12	10	8		
$d_3(x)$					8,9	9	9		
$f_2(x)$		19	17	13					
$d_2(x)$		6	6	6					
$f_1(x)$	16								
$d_1(x)$	4								

четвертому етапі, то

$$f_4(8) = 8,\ d_4(8) = 10,\ f_9(4) = 4,\ d_4(9) = 10.$$

Переходимо до третього етапу. Якщо ми перебуваємо у вершині 5, то з неї існує два можливі шляхи до вершини 10. Очевидно, що

$$f_3(5) = \min\{4 + f_4(8), 8 + f_4(9)\} = \min\{12, 12\} = 12,$$
$$d_3(5) = 8 \text{ або } 9.$$

Аналогічно знаходимо всі величини $f_k(x)$, $d_k(x)$. Результати подані в табл. 3.8.

Як видно з табл., довжина найкоротшого шляху з вершини 1 до вершини 10 дорівнює 16, а оптимальним маршрутом є $1 \to 4 \to 6 \to 9 \to 10$. Цей маршрут будуємо згідно з $d_k(x)$

$$1 \to d_1(1) = 4 \to d_2(4) = 6 \to d_3(6) = 9 \to d_4(9) = 10.$$

Зауважимо таке: якщо c_{ij} — собівартість перевезення вантажу відповідною комунікацією, то розглянута задача еквівалентна пошуку шляху мінімальної собівартості з вершини 1 до вершини 10.

Задачі

3.1. Нехай 20 умовних одиниць ресурсу можуть бути інвестовані в розвиток трьох галузей промисловості. Прибутки, які отримують від інвестування x одиниць ресурсу в кожну з галузей, відповідно дорівнюють

$$g_1(x) = 5\sqrt{(x)},\ g_2(x) = x,\ g_3(x) = 0.07x^2,\ 0 \leqslant x \leqslant 20.$$

Ресурс розподіляється в цілих одиницях. Визначити кількість одиниць ресурсу, що інвестується в кожну з галузей так, щоб отримати максимальний сумарний прибуток. Знайти величину цього прибутку.

3.2. Підприємство, що виготовляє новий пральний порошок, зацікавлене в оптимальних інвестиціях в різні засоби реклами, щоб максимізувати свій прибуток від реалізації продукції. Розглядають чотири засоби реклами: газета (г), журнал (ж), телебачення (т), радіо (р). В табл. 3.9 подано очікувані прибутки в умовних одиницях від інвестування в різні засоби реклами. На рекламу виділено 10 000 одиниць. В які засоби реклами варто їх інвестувати, щоб максимізувати загальний прибуток. Розв'язати цю задачу, якщо на рекламу виділено 5 000 одиниць.

3.3. 1) На графі, зображеному на рис. 1, знайти найкоротший шлях з вершини 1 до вершини 10, що проходить через вершину 8.

2) Модифікувати алгоритм розв'язання задачі про найкоротший шлях на мережі на випадок задачі про знаходження найдовшого шляху між двома заданими вершинами на мережі. Знайти найдовший шлях з вершини 1 до вершини 10 на графі, зображеному на рис. 1.

3.4. Вантажний автомобіль треба перегнати з пункту 1 до пункту 12. Дорогою він може перевозити вантажі та принести прибутки, величини яких зазначені в табл. 3.10. Зауважимо, що з заданого пункту можна проїхати лише до деяких

Табл. 3.9.

Інвестиція, тис. од.	Прибуток тис. од.			
	г	ж	т	р
0	10.00	10.00	10.00	10.00
1	10.20	10.25	10.30	10.15
2	10.70	10.80	10.90	10.50
3	11.20	11.65	11.95	11.20
4	12.00	12.60	13.00	12.00
5	12.65	13.75	14.50	13.10
6	13.30	14.80	16.30	14.50
7	14.06	15.95	18.05	15.60
8	14.80	17.20	20.00	17.20
9	15.40	18.10	21.70	18.64
10	16.00	19.00	24.00	20.00

інших пунктів. Наприклад, з пункту 2 можна безпосередньо проїхати до пунктів 5, 6, 7 та 8. Потрібно визначити маршрут перегону автомобіля, що максимізує загальний прибуток.

3.5. Використовуючи концепцію динамічного програмування, визначити шлях максимальної довжини з вершини 1 до вершини 7 на графі, зображеному на рис. 2.

3.6. Використовуючи алгоритм розв'язування задачі планування виробництва та запасів, розв'язати приклад з пункту 1.4 за умови, що початковий запас продукції становить 9 одиниць, а запас продукції не може перевищувати 10 одиниць.

3.7. 1) Розглянути задачу згладжування виробництва з 10-ма періодами, якщо початковий запас становить 15 одиниць, максимальний запас продукції на кінець кожного місяця — 20 одиниць, вартість виробництва j одиниць продукції

Табл. 3.10.

Від пункту	До пункту										
	2	3	4	5	6	7	8	9	10	11	12
1	5	4	2								
2				8	10	5	7				
3				6	3	8	10				
4				8	9	6	4				
5								8	4	3	
6								5	2	7	
7								4	10	6	
8								12	5	2	
9											7
10											3
11											6

протягом кожного періоду — $25 + 6j$, вартість зберігання j одиниць продукції на кінець кожного періоду становить $\frac{j}{2}$, а об'єми попиту подано в таблиці: Використовуючи алго-

Табл. 3.11.

Період i	1	2	3	4	5	6	7	8	9	10
Період d_i	15	12	30	25	24	6	18	25	12	40

ритм пункту 1.4, визначити план виробництва на кожний період, що мінімізує сумарні витрати.

2) Розглянути задачу згладжування виробництва, об'єми попиту якої подано в табл. 3.11, вартість налагодження виробництва — 5 одиниць, вартість виробництва одного виробу - 1 одиниця, вартість зберігання одного виробу - 1

одиниця і визначається запасом на кінець місяця. Запас на

Табл. 3.12.

Місяць	Червень	Липень	Серпень
Попит	5	3	2

1-ше червня становить 1 одиницю, а на кінець серпня – 0. Побудувати функціональне рівняння динамічного програмування для визначення оптимального плану виробництва.

3.8. Розв'язати задачу про заміну обладнання, наведену в пункті 1.3, в якій величини всіх прибутків збільшені на 2 одиниці, всі витрати на утримання збільшені на 3 одиниці і всі вартості заміни збільшені на 1 одиницю.

3.9. Методом динамічного програмування розв'язати задачу про оптимальний розподіл капіталовкладень з прикладу 2.10 при довільних значеннях $I \in 1, ..., 10$.

3.10. Нехай у задачі цілочислового програмування (3.2) функції $p_j(x_j)$ задані у вигляді таблиці. Порівняти кількість арифметичних операцій та операцій порівняння чисел у методі динамічного програмування і при використанні методу повного перебору всіх допустимих розв'язків.

Розділ 4

Потоки в мережі

4.1 Задача про найкоротший шлях

Нагадаємо необхідні означення.

Означення 4.1. *Орієнтований граф складається з множини вершин $V = \{1, ..., n\}$ і множини E впорядкованих пар різних вершин. Кожна така пара $(i, j) \in E$ називається дугою.* ***Мережею називається орієнтований граф****, кожній дузі $(i, j) \in E$ якого поставлено у відповідність невід'ємне число b_{ij}. Ці числа можуть визначати довжину, час, пропускну здатність і т.д.*

Ми будемо розглядати мережі, в яких вершина 1 виокремлена як початкова, а вершина n — як кінцева[1]. Всі інші вершини $i \neq 1, n$ будемо називати *проміжними*.

Якщо $(i, j) \in E$, то вважатимемо, що вершина i *передує* вершині j, а вершина j *наступна* за вершиною i.

Нехай $P(i) = \{k \in V : (k, i) \in E\}$ — множина вершин k, які передують вершині i, а $D(i) = \{j \in V : (i, j) \in E\}$ — множина вершин j, наступних за вершиною i.

Надалі кожне число b_{ij} будемо інтерпретувати як довжину відповідної дуги $(i, j) \in E$.

[1]Вершини в мережі також називають вузлами.

Шляхом, прокладеним із вершини i_1 у вершину i_k, називається переміжна послідовність вершин і дуг мережі вигляду

$$i_1, (i_1, i_2), i_2, (i_2, i_3), i_3, ..., (i_{k-2}, i_{k-1}), i_{k-1}, (i_{k-1}, i_k), i_k, \qquad (4.1)$$

в якій всі вершини $i_1, ..., i_k$ різні.

Надалі будемо говорити про шлях з i_1 в i_k і позначати його так: $i_1 \to i_2 \to ... \to i_k$. Вершини $i_2, ..., i_{k-1}$ називаються *проміжними*. Довжина шляху $i_1 \to i_2 \to ... \to i_k$ дорівнює сумі довжин дуг, які його становлять

$$\sum_{l=1}^{k-1} b_{i_l i_{l+1}}.$$

Послідовність дуг і вершин вигляду (4.1) називається *циклом*, якщо в ній всі проміжні вершини різні, а $i_1 = i_k$.

Задача про найкоротший шлях: *треба знайти шлях мінімальної довжини, що веде з вершини 1 у вершину n, тобто визначити в транспортній мережі найкоротший шлях між заданим вихідним пунктам і пунктом призначення.*

Як приклад розглянемо задачу про найкоротший шлях між двома містами 1 і n. Для визначення мережі потрібно задати її дуги. Враховуючи зміст задачі, дорогу довжини b_{1j} між містом 1 і будь-яким сусіднім містом j вважатимемо такою, що виходить з міста 1. Зіставимо їй дугу $(1, j)$, яка виходить з вершини 1. Аналогічно дорогу довжини b_{in}, що зв'язує місто n з будь-яким сусіднім містом i, будемо вважати такою, що входить в місто n. Зіставимо їй дугу (i, n), яка входить у вершину n. Розглянемо дорогу довжини b_{ij}, що з'єднує міста $i, j \neq 1, n$. Якщо по ній можна проїхати у двох напрямах (що зазвичай і буває), то в мережі вводяться дві дуги (i, j) і (j, i), причому $b_{ij} = b_{ji}$. Задачу про найкоротший шлях з міста 1 в місто 4 ілюструє рис. 4.1.1. Числа в дужках позначають довжини доріг у кілометрах.

Найпоширеніші два алгоритми для розв'язання задачі пошуку найкоротшого шляху в мережах, які мають цикли, та в мережах без циклів.

1. Алгоритм Дейкстри.

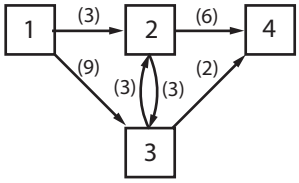


Рис. 4.1.1.

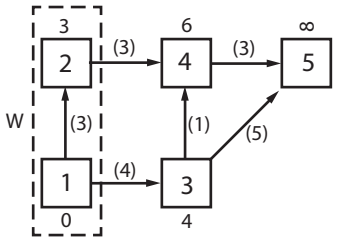


Рис. 4.1.2.

2. Алгоритм Флойда.

Алгоритм Дейкстри розроблений для пошуку найкоротшого шляху між заданою вихідною вершиною та будь-якою іншою вершиною мережі. *Алгоритм Флойда* більш загальний, оскільки він дає змогу одночасно знайти найменші шляхи між будь-якими двома вершинами мережі.

Алгоритм Дейкстри розв'язування задачі про найкоротший шлях

У процесі реалізації алгоритму Дейкстри при переході від вершини i до наступної вершини j використовують спеціальну процедуру позначки дуг.

На кожному кроці алгоритму є множина вершин W, яка містить вершину 1, і позначки $\rho(j)$, які визначені для всіх вершин $j \in V$. За змістом $\rho(j)$ — довжина найкоротшого шляху з вершини 1 у вершину j, в якому всі проміжні вершини належать множині W. Якщо для деякої вершини j, зазначеного шляху не існує, то приймають $\rho(j) = \infty$.

Покажемо на прикладі, що довжина найкоротшого шляху з 1 в j може бути менше $\rho(j)$. На рис. 4.1.2 позначки $\rho(j)$ розташовують знизу або зверху від вершин j, а поряд із кожною дугою (i, j) в дужках зазначена її довжина b_{ij}. На рис. 4.1.1 $W = \{1, 2\}$, $\rho(4) = 6$ і шлях $1 \to 2 \to 4$ довжиною 6 веде з вершини 1 у вершину 4 через вершину $2 \in W$. Позаяк найкоротший шлях з 1 в 4 є шлях $1 \to 3 \to 4$ довжини 5. Зауважимо, що $\rho(5) = \infty$, оскіль-

ки не існує шляху з вершини 1 у вершину 5, який має проміжні вершини, що належать множині W.

Розглянемо реалізацію алгоритму за кроками.

Крок 1. Приймемо $W = \{1\}$, $\rho(j) = \begin{cases} b_{1j}, & (1,j) \in E, \\ \infty, & (1,j) \notin E. \end{cases}$ Нехай після k кроків є деяка множина W і значення позначок $\rho(j),\ j \in V$.

Крок $k+1$. Вершина

$$i \in Arg \min_{j \notin W} \rho(j) \tag{4.2}$$

додається до множини W, а для вершин $j \in D(i)$ значення позначок $\rho(j)$ перераховуєть за формулою

$$\rho(j) := \min\{\rho(j), \rho(i) + b_{ij}\}. \tag{4.3}$$

Реалізація алгоритму закінчується, коли кінцева вершина n буде входити до множини W. У цьому разі найкоротший шлях з 1 в n довжиною $\rho(n)$ відновлюється так: спочатку знаходимо вершину $k \in P(n)$ з умови $\rho(n) = b_{kn} + \rho(k)$, потім вершину $s \in P(k)$ з умови $\rho(k) = b_{sk} + \rho(s)$ і т. д.

Розглянемо обґрунтування алгоритму Дейкстри.

Лема 4.1. *Нехай* Π *— найкоротший шлях з вершини 1 у вершину* $i \notin W$*, який визначається за (4.2). Тоді всі проміжні вершини шляху* Π *належать множині* W.

Доведення. Справді, припустимо протилежне. Рухаючись уздовж шляху Π, починаючи від вершини 1, знайдемо першу вершину $k \notin W$. Тоді за припущенням $k \neq i$ і позначка $\rho(k)$ менша від довжини шляху π, яка, зокрема, не перевищує позначки $\rho(i)$. Отже, $\rho(k) < \rho(i)$, що суперечить вибору вершини i. □

Лема 4.2. *Нехай вершина* i *задовольняє умову (4.2) і приєднана до множини* W *(тобто* $W := W \cup \{i\}$*), а позначки* $\rho(j)$*, де* $j \in D(i)$*, перераховані за формулою (4.3). Тоді нові позначки* $\rho(j)$ *відповідають їхнім визначенням:* $\rho(j)$ *— довжина найкоротшого шляху з 1 в* j*, що має проміжні вершини, які належать* W.

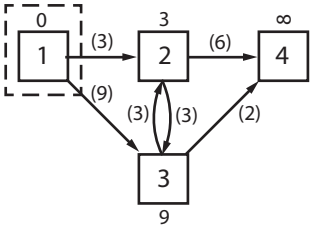


Рис. 4.1.3.

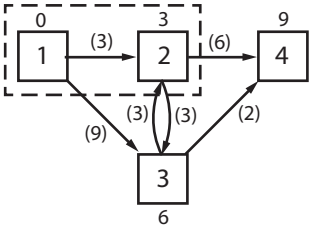


Рис. 4.1.4.

Доведення. Виберемо довільну вершину $j \in D(i)$. Нехай виконується нерівність $\rho(j) \leqslant \rho(i) + b_{ij}$. Тоді найкоротший шлях з 1 в j довжиною $\rho(j)$ з проміжними вершинами, які належать множині W, при додаванні вершини i не зміниться. Отож, у цьому випадку позначка $\rho(j)$ відповідає своєму визначенню.

Припустимо тепер, що виконується нерівність $\rho(j) > \rho(i)+b_{ij}$. Множина W містить вершину i. Тому найкоротший шлях з 1 в j з проміжними вершинами з W буде йти спочатку з 1 в i, а потім — по дузі (i, j). І в цьому випадку позначка $\rho(j)$ відповідає своєму визначенню. □

Зауваження 4.1. Зазначимо, що після того як на деякому кроці алгоритму Дейкстри вершина i добавлена в множину W, на наступних кроках позначка $\rho(i)$ не змінюється і $\rho(i)$ — довжина найкоротшого шляху з 1 в i. Отже, на кожному кроці алгоритму позначки $\rho(j)$ можна перераховувати тільки для вершин $j \notin W$. Якщо сформульована задача визначення мінімальних шляхів з вершини 1 у кожну вершину $j \in V$, то алгоритм треба зупинити після того, як множина W збігається з множиною V.

Приклад 4.1. *Знайдемо найкоротший шлях з міста 1 до міста 4 для мережі, зображеної на рис. 4.1.1. На рис. 4.1.3 – 4.1.6 штриховими лініями відзначені множини W та позначки вершин для чотирьох кроків алгоритму Дейкстри. Найкоротший шлях* $1 \to 2 \to 3 \to 4$. ▲

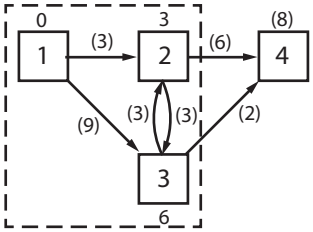


Рис. 4.1.5.

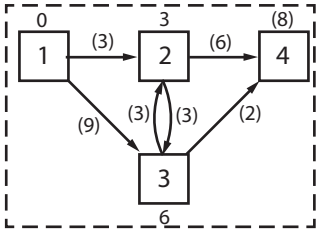


Рис. 4.1.6.

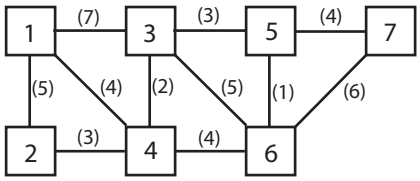


Рис. 4.1.7.

Задачі

4.1. Знайти найкоротший шлях з вершини 1 у вершину 5 для мережі, зображеної на рис. 4.1.2.

4.2. Знайти всі найкоротші шляхи з міста 1 в місто 7 для мережі доріг, що зображені на рис. 4.1.7.

4.3. Довести, що в алгоритмі Дейкстри після додавання вершини i в множину W помітка $\rho(i)$ не змінює свого значення до завершення алгоритму.

4.4. Чи може після завершення алгоритму Дейкстри деяка вершина j не належати множині W? Якщо може, то навести приклад.

4.2 Мережеве планування

Організовуючи роботи над різними проектами (інвестиційні проекти, комплексні наукові дослідження і т.д.), набули поширення методи мережевого планування. Розглянемо їхню головну ідею.

Будемо говорити, що робота k безпосередньо передує роботі l, якщо в момент завершення роботи k і, можливо, деяких інших робіт може розпочатися виконання роботи l.

Початкова інформація про проект задається переліком робіт, їхньою тривалістю та черговістю виконання. Черговість виконання означає, що для кожної роботи має бути зазначено, яким роботам вона безпосередньо передує.

Для наочності проект зображають *мережевим графіком.*

Означення 4.2. ***Мережним графіком*** *називається мережа без циклів з множиною вершин $V = \{1, ..., n\}$ і множиною дуг E, в якій напрямлені дуги відповідають роботам, а вершини – подіям, які відбуваються під час завершення деяких робіт. Ніяка робота, що виходить з деякої вершини, не може початися раніше, ніж завершаться всі роботи, що входять в цю вершину. У мережі виділено дві вершини: початкова та кінцева, які відповідають моментам початку і закінчення всіх робіт проекту. У початкову вершину дуги не можуть входити, а з кінцевої вершини вони не можуть виходити. Інші вершини мають як вхідні, так і вихідять дуги. Без втрати загальності будемо вважати, що 1 — початкова, а n — кінцева вершини. Кожній дузі (роботі) $(i, j) \in E$ мережевого графіка зіставлена тривалість її виконання t_{ij}.*

Табл. 4.1.

Найменування роботи	Безпосередньо передує роботам	Час виконання, місяців
1. Підбір акціонерів	3,4	4
2. Пошук приміщення	5,6	5
3. Внески акціонерів	5,6	3
4. Ліцензування	7	4
5. Реєстрація	7	3
6. Обладнання офісу	-	6
7. Наймання працівників	-	2

Приклад 4.2. Підприємець вирішив організувати фірму. У табл. 4.1 подано інформацію про роботи, які необхідні для створення фірми. У другому стовпці зазначені роботи, яким ця робота безпосередньо передує. Мережевий графік будуємо, починаючи з кінцевої вершини. У неї входять роботи з номерами 6 і 7. Роботі 7 передують роботи 4 і 5.

Треба ввести подію, яка полягає в закінченні робіт 4 та 5 і т.д. На рис. 4.2.8 зображено відповідний мережевий графік. На дугах зазначено номери робіт, а в дужках їхня тривалість у місяцях. ▲

Зауваження 4.2. У вихідній інформації з проекту можуть міститися помилки. Нехай у прикладі 4.2 помилково зазначено, що перша робота безпосередньо передує п'ятій, тобто робота 1 передує роботам із списку 3,4,5. Як виявити і виправити подібні помилки? Послідовно беремо роботи з зазначеного списку і знаходимо всі наступні за ними роботи. Серед подальших робіт не повинна траплятися робота зі списку 3,4,5. Якщо ж така трапляється (в цьому випадку п'ята), то її треба вилучити зі списку.

Зауваження 4.3. Деякі роботи можуть виконуватися одночасно. Тоді виникне мультиграф, тобто деякі вершини будуть з'єднані декількома дугами (рис. 4.2.9).

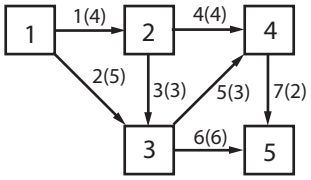


Рис. 4.2.8.

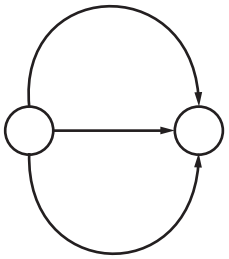


Рис. 4.2.9.

Щоб отримати звичайний орієнтований граф, потрібно на кожній дузі ввести подію і фіктивну роботу нульової тривалості.

Пронумеруємо вершини, присвоївши початковій вершині номер 1, а кінцевій — номер n. Будемо говорити, що вершина i передує вершині j, якщо існує дуга, що виходить з вершини i та входить у вершину j.

Уведемо поняття *рангу* вершини. Початковій вершині присвоюємо нульовий ранг. До вершин першого рангу зачислено ті, яким передує лише вершина нульового рангу та ніякі інші. Вершини рангу k — це ті вершини, яким передують тільки вершини рангу не вище $k-1$.

Лема 4.3. *Методом присвоєння рангу можна визначити ранги всіх вершин мережевого графіка.*

Доведення. Справді, припустимо протилежне і вважатимемо, що ми визначили ранги вершин з деякої множини W, але вершинам з доповнювальної множини $V \setminus W$ ранг приписати не можна. Візьмемо будь-яку вершину без рангу $i_1 \notin W$. За припущенням їй не можна приписати ранг. Це означає, що їй передує вершина без рангу $i_2 \notin W$, якій також передує ще одна вершина без рангу $i_3 \notin W$ і т.д. Оскільки мережевий графік містить скінченну кількість вершин, то, зрештою, за деякого k знайдеться вершина $i_l \in \{i_1, ..., i_{k-1}\}$, яка передує вершині i_k. У підсумку отримаємо

цикл $i_k \to i_{k-1} \to ... \to i_l \to i_k$, який суперечить означенню мережевого графіка. □

Розглянемо *алгоритм знаходження рангів вершин мережевого графіка*. Для цього припишемо кожній дузі $(i, j) \in E$ довжину $b_{ij} = 1$. Довжина шляху з вершини 1 у вершину j дорівнює кількості дуг, з яких він складається. Максимальний шлях з вершини 1 у вершину j має найбільшу довжину серед усіх шляхів, що ведуть з 1 в j. Неважко довести, що довжина максимального шляху з 1 в j дорівнює рангу $r(j)$ вершини j. Тому для визначення рангів вершин скористаємося алгоритмом Дейкстри, модифікованим для пошуку довжин максимальних шляхів із вершини 1 у будь-яку вершину $j \in V$.

Крок 1. Приймемо $W = \{1\}$,

$$\rho(j) = \begin{cases} 1, & (1, j) \in E, \\ 0, & (1, j) \notin E. \end{cases}$$

Нехай після k кроків є деяка множина W і значення позначок $\rho(j), j \in V$.

Крок $k + 1$. Вершина

$$i \in Arg \max_{j \notin W} \rho(j) \tag{4.4}$$

додається до множини W, а для вершин $j \in D(i)$ значення позначок $\rho(j)$ перераховуються за формулою

$$\rho(j) := \max\{\rho(j),\, \rho(j) + b_{ij}\}. \tag{4.5}$$

Алгоритм закінчить свою роботу, коли множина W збігається з множиною V. Підсумкові значення позначок $\rho(j)$ будуть дорівнювати рангам вершин $r(j)$.

Обґрунтування цього алгоритму майже подібне до обґрунтування алгоритму пошуку мінімального шляху з вершини 1 у вершину j.

Після того як знайдені ранги всіх вершин мережевого графіка, вершини зручно перенумерувати відповідно до такого правила. Вершинам рангу 1 присвоюємо номери $2, ..., l$, вершинам

рангу 2 — номери $l+1, ..., s$ і т.д. Нарешті, n — номер кінцевої вершини. Мережевий графік за зазначеним способом нумерації вершин називається *правильно пронумерованим*. Для правильно пронумерованого мережевого графіка алгоритми обчислення його характеристик значно спрощуються.

Довжиною шляху мережного графіка з вершини (події) 1 у вершину (подію) j будемо називати суму тривалостей робіт, які його становлять. Виникає запитання про мінімальний час t_j реалізації якої-небудь події j. Для початкової події 1 приймемо $t_1 = 0$, тобто проект починається в нульовий момент часу. Нагадаємо, що через $P(j)$ позначаємо множину всіх вершин, які передують вершині j.

Лема 4.4. *Мінімальний час t_j виконання події j дорівнює довжині найбільш тривалого шляху, що веде з вершини 1 у вершину j.*

Доведення. Доведемо, використовуючи метод математичної індукції за рангом вершини j. Для вершини 1 ранг дорівнює нулю і твердження очевидне. Нехай для всіх вершин i рангу не вище $k-1$ твердження леми правильне. Візьмемо вершину j рангу k. Нехай t – довжина найдовшого шляху, який веде з вершини 1 у вершину j. Покажемо, що $t_j = t$. Неважко бачити, що $t_j \geqslant t$. Справді, подія j не може відбутися, якщо не виконані всі роботи, які належать найдовшому шляху з 1 в j. Далі всі вершини $i \in P(j)$ мають ранг не вище $k-1$. Отже, за припущенням індукції t_i – довжина найдовшого шляху, що веде з вершини 1 у вершину i. Тому

$$t = \max_{i \in P(j)} (t_i + t_{ij}).$$

Для виконання події j за час t достатньо, щоб кожна подія $i \in P(j)$ була виконана за час t_i і за нею негайно починалася робота (i, j). Отже, $t_j = t$. □

З леми 4.4 безпосередньо отримуємо таку процедуру пошуку величин t_j, яка називається **алгоритмом Форда**.

Для вершин j першого рангу очевидно, що $t_j = t_{1j}$. Нехай знайдені величини t_i для усіх вершин i рангу не вище $k-1$. Тоді для вершини рангу k отримаємо

$$t_j = \max_{i \in P(j)} (t_i + t_{ij}). \tag{4.6}$$

У підсумку для кінцевої вершини (події) n знаходимо величину t_n, яку називають *критичним часом* проекту.

Означення 4.3. ***Критичним шляхом*** *називається шлях довжиною t_n, який веде з початкової вершини до кінцевої.*

Для знаходження критичного шляху достатньо, рухаючись з вершини n, спочатку знайти вершину $s \in P(n)$ з умови $t_n = t_s + t_{sn}$, потім вершину $l \in P(s)$ з умови $t_s = t_l + t_{ls}$ і т.д.

Зазначимо, що алгоритм Форда використовує ранги вершин і правильну нумерацію мережевого графіка. Якщо ранги вершин невідомі, то для визначення часу t_j можна використовувати алгоритм Дейкстри. Перейдемо до визначення інших характеристик мережевого графіка.

Означення 4.4. ***Максимальним часом*** T_i *події i називається найбільший час її реалізації, за якого проект можна виконати за критичний час t_n.*

Розглянемо алгоритм знаходження часів T_i. Нагадаємо, що для вершини i через $D(i)$ позначається множина вершин j, яким передує вершина i.

Очевидно, що $T_n = t_n$. Нехай визначені числа $T_n, T_{n-1}, ..., T_{i+1}$.

Лема 4.5. *Для кожного $j \in D(i)$, час T_j, пов'язаний з часом T_i співвідношенням*

$$T_i = \min_{j \in D(i)} (T_j - t_{ij}). \tag{4.7}$$

Доведення. Нехай $s \in D(i)$ — номер вершини, на якому досягається мінімум у формулі (4.7). Припустимо, що $T_i > T_s - t_{is}$. Якщо подія i виконана в момент T_i, то за визначенням часу T_i

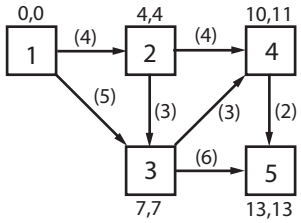


Рис. 4.2.10.

проект може бути виконаний за критичний час. З останньої нерівності випливає, що подію s не можна виконати за час T_s і за визначенням T_s, час виконання всього проекту буде більшим за t_n (суперечність). Отже, $T_i \leqslant T_s - t_{is}$. З іншого боку, якщо подія i виконана в момент $T_s - t_{is}$, то кожна подія $j \in D(i)$ може бути виконана не пізніше моменту $T_s - t_{is} + t_{ij} \leqslant T_j$ і проект можна виконати за критичний час. Отже, $T_i = T_s - t_{is}$. □

З леми 4.5 випливає, що всі часи $T_n, T_{n-1}, ..., T_1$ можна послідовно знайти за формулою (4.7). Для подій i, що входять до критичного шляху, $T_i = t_i$. Обчислення величин t_j, T_j дає змогу визначити резерви часу для кожної роботи.

Означення 4.5. *Величина $T_j - t_i - t_{ij}$ називається повним резервом часу роботи (i, j). Це максимально можливе збільшення тривалості роботи (i, j) без збільшення тривалості виконання всього проекту. Вільний резерв часу $(t_j - t_i - t_{ij})$ — це максимально можливе збільшення тривалості роботи (i, j), за якого можна розпочати всі роботи, які виходять з вершини j в найбільш ранній час. Незалежний резерв часу* $\max\{0,\, t_j - T_i - t_{ij}\}$ *— це максимально можливе збільшення тривалості роботи (i, j), яке не впливає на резерв часу інших робіт.*

Визначення незалежного резерву часу роботи потребує деякого пояснення. Розглянемо три послідовні події k, i, j, з'єднані роботами (k, i) і (i, j). Нехай виконується нерівність $T_i > t_i$ і незалежний резерв роботи (i, j) додатний, тобто $t_j - T_i - t_{ij} > 0$.

Припустимо, що незалежний резерв роботи (i, j) використовується повністю. Тоді тривалість цієї роботи збільшиться до величини $t_j - T_i$. Якщо почати її в момент T_i і закінчити в момент t_j, то резерви інших робіт не зменшаться. Наприклад, можна використовувати повний резерв роботи (k, i), почавши її в момент t_k і закінчивши в момент T_i.

У прикладі 4.2 з організацією фірми знайдемо всі наведені характеристики проекту. На рис. 4.2.10 над (або під) вершинами зазначено пари t_j, T_j.

Критичний шлях: $1 \to 2 \to 3 \to 5$. Критичний час виконання всього проекту – 13 місяців. Повний резерв часу роботи $(2, 4)$ (реєстрація) дорівнює 3 місяці, а аналогічний резерв для роботи $(3, 4)$ (наймання працівників) становить 1 місяць. Вільний і незалежний резерви роботи $(2, 4)$ становлять по 2 місяці кожний.

Задачі

4.5. Індукцією за величиною рангу довести, що ранг $r(j)$ вершини j дорівнює довжині максимального шляху з вершини 1 до вершини j.

4.6. Використовуючи алгоритм Дейкстри, знайти ранги вершин і правильно пронумерувати мережевий графік, зображений на рис. 4.2.11.

4.7. Довести, що правильно прономерована мережа не містить циклів.

4.8. За якого рангу кінцевої вершини n можлива єдино правильна нумерація мережевого графіка?

4.9. Довести, що для будь-якої вершини i критичного шляху виконується рівність $T_i = t_i$.

4.10. Знайти критичний шлях, критичний час проекту, а також резерви робіт мережевого графіка, зображеного на рис. 4.2.11.

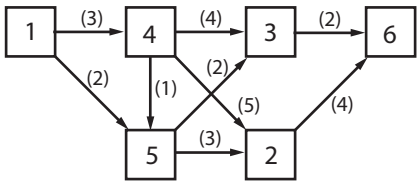


Рис. 4.2.11.

4.3 Задача про максимальний потік

У цьому параграфі розглянемо задачу максимізації потоку деякого продукту мережею. Подібні задачі виникають при водопостачанні міст, перекачуванні нафти і т.д. Вивчають також інформаційні потоки в мережах.

Уточнимо, що розумітимемо під потоком продукту, який рухається по деякій дузі мережі. Нехай як дуга мережі буде труба водопроводу. Потік води по трубі вимірюється кількістю рідини, що протікає в одиницю часу через поперечний переріз труби. Напрям руху води має зазвичай сталу орієнтацію, яка зобумовлена розташуванням у мережі насосів або гідравлічних веж. Максимальна величина потоку по трубі залежить від її діаметра, потужності насосів або висоти гідравлічних веж.

Означення 4.6. *Потоковою мережею називається орієнтований граф з множиною вершин* $V = \{1, ..., n\}$ *і множиною дуг* E*, що задовольняє такі умови. У графа виділено дві вершини: початкова та кінцева. З початкової вершини продукт по дугах мережі направляється до кінцевої. Кожній дузі* $(i, j) \in E$ *зіставлять число* b_{ij}*, яке називаємо пропускною здатністю. Потік продукту вздовж дуги може переміщатися тільки в додатному напрямі, тобто від вершини* i *до вершини* j*, і його величина не перевищує* b_{ij}*.*

Без втрати загальності припустимо, що початкова вершина мережі має номер 1, а кінцева — номер n. Для проміжних вершин i та j можливий випадок, коли їх з'єднують дві дуги в про-

тилежних напрямах, тобто одночасно $(i,j) \in E$ та $(j,i) \in E$. Пропускні здатності цих дуг можуть бути різними. Наприклад, потік транспорту вздовж магістралі рухається, зазвичай, у двох напрямах.

Нагадаємо, що для вершини мережі i через $D(i)$ позначаємо множину всіх вершин $j \in V$, які наступні за вершиною $i : (i,j) \in E$, а через $P(i)$ — множину всіх вершин k, які передують вершині $i : (k,i) \in E$.

Означення 4.7. *Потоком мережею (або просто потоком) називається вектор вигляду $x = (x_{ij}, (i,j) \in E)$. Потік x називається допустимим, якщо він задовольняє обмеження*

$$\sum_{j \in D(i)} x_{ij} - \sum_{k \in P(i)} x_{ki} = 0, \quad i \in V \setminus \{1, n\}, \tag{4.8}$$

$$0 \leqslant x_{ij} \leqslant b_{ij}, \quad (i,j) \in E.$$

Зокрема, потік продукту дугою (i,j) дорівнює x_{ij}. Сума

$$v = \sum_{j \in D(1)} x_{1j}$$

називається величиною потоку. За змістом вона визначає загальну кількість продукту, що виходить з початкової вершини в одиницю часу.

Зауважимо, що рівності (4.8), які виконуються в проміжних вершинах $i \in V \setminus \{1, n\}$ мережі, відображають *закон збереження потоку*. Якщо в цих вершинах відбувається споживання продукту, то закон збереження порушується.

Величину потоку v можна задати також у вигляді

$$v = \sum_{k \in P(n)} x_{kn}. \tag{4.9}$$

Справді, склавши рівності (4.8), отримаємо

$$-\sum_{j \in D(1)} x_{1j} + \sum_{k \in P(n)} x_{kn} = 0,$$

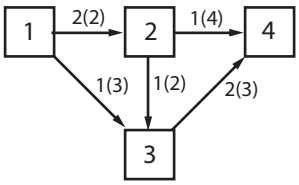


Рис. 4.3.12.

оскільки кожний доданок x_{ij}, де $i \neq 1$, $j \neq n$, в отриману суму входить двічі: зі знаками плюс і мінус.

На рис. 4.3.12 зображена мережа з множиною вершин $V = \{1, 2, 3, 4\}$ і множиною дуг $E = \{(1,2), (1,3), (2,3), (2,4), (3,4)\}$. Числа в дужках, які розташовані поряд із дугами, визначають їхню пропускну здатність, а числа без дужок — компоненти потоку $x = (2, 1, 1, 1, 2)$ величини 3.

Розглянемо задачу про максимальний потік

$$
\begin{aligned}
&\max v,\\
\text{за умов}\quad &\sum_{j\in D(1)} x_{1j} - v = 0,\\
&\sum_{j\in D(i)} x_{ij} - \sum_{k\in P(i)} x_{ki} = 0, \quad i \in V \setminus \{1, n\},\\
&v - \sum_{k\in P(n)} x_{kn} = 0,\\
&0 \leqslant x_{ij} \leqslant b_{ij}, \quad (i,j) \in E.
\end{aligned}
\tag{4.10}
$$

Задача (4.10) — це задача LP зі змінними x_{ij}, $(i,j) \in E$, і v. Легко бачити, що нульовий потік допустимий. Отож, обмеження задачі (4.10) сумісні. Запишемо двоїсту задачу. Для цього першим n рівностям зіставимо двоїсті змінні y_i, а нерівностям $x_{ij} \leqslant b_{ij}$ —

двоїсті змінні z_{ij}, $(i,j) \in E$. Двоїста задача LP набуде вигляду

$$\begin{aligned} &\min \sum_{(i,j)\in E} b_{ij} z_{ij}, \\ &\text{за умов} \quad y_i - y_j + z_{ij} \geqslant 0, \;\; (i,j) \in E, \\ &\qquad\quad -y_1 + y_n \geqslant 1, \;\; z_{ij} \geqslant 0, \;\; (i,j) \in E. \end{aligned} \tag{4.11}$$

Де нерівність $-y_1 + y_n \geqslant 1$ відповідає змінній v прямої задачі (4.10). Змінні y_i, $i = 1, ..., n$, — вільні, оскільки вони відповідають обмеженням-рівностям задачі (4.10).

Задачі

4.14. Довести, що задача про максимальний потік завжди має оптимальний розв'язок.

4.15. Розглянемо потокову мережу зі споживанням продукту. Нехай його споживання за одиницю часу у проміжній вершині i становить сталу величину d_i, $i = 2, ..., n-1$.

1) Виписати формулу, аналогічну до (4.9), для величини потоку.

2) Сформулювати задачу про максимальний потік. Записати двоїсту задачу.

4.4 Пошук максимального потоку

Запишемо величину максимального потоку через пропускні здатності дуг мережі та сформулюємо алгоритм його пошуку.

Означення 4.8. *Перерізом називається розбиття* $(W, \overline{W})$ *множини вершин мережі* V *на такі дві диз'юнктивні підмножини* W *і* $\overline{W}$, *що* $1 \in W$, $n \in \overline{W}$. *Дуга* (i,j) *називається* ***прямою дугою перерізу*** $(W, \overline{W})$, *якщо* $i \in W$, $j \in \overline{W}$. *Для* ***зворотної дуги перерізу*** *виконується умова* $i \in \overline{W}$, $j \in W$. ***Пропускною здатністю перерізу*** *називається сума пропускних здатностей всіх*

його прямих дуг

$$C(W,\overline{W}) = \sum_{(i,j)\in E:\, i\in W, j\in\overline{W}} b_{ij}.$$

Переріз, що має найменшу пропускну здатність, називається ***мінімальним****.*

Приклад 4.3. Розглянемо мережу на рис. 4.3.12. Переріз $(W,\overline{W}) = (\{1,3\},\{2,4\})$ має прямі дуги $(1,2)$, $(3,4)$ і зворотну дугу $(2,3)$. Величина перерізу дорівнює $C(W,\overline{W}) = 5$. ▲

Лема 4.6. *Кожний переріз $(W,\overline{W})$ визначає допустимий розв'язок вартості $C(W,\overline{W})$ для задачі (4.11), яка двоїста до задачі про максимальний потік, за формулою*

$$z_{ij} = \begin{cases} 1, & i\in W, j\in\overline{W}, \\ 0, & \textit{інакше}, \end{cases} \qquad y_i = \begin{cases} 0, & i\in W, \\ 1, & i\in\overline{W}. \end{cases} \tag{4.12}$$

Доведення. Для розв'язку вигляду (4.12) перевіримо виконання обмеження задачі (4.11). Розглянемо чотири випадки, оскільки вершини i та j можуть належати або не належати до множин W і $\overline{W}$. У будь-якому випадку відповідну нерівність $y_i - y_j + z_{ij} \geqslant 0$ легко перевірити, зокрема, вона буде строга тоді і лише тоді, коли (i,j) є зворотна дуга перерізу $(W,\overline{W})$. Далі за означенням розрізу $1\in W$, $n\in\overline{W}$. Тому $y_1 = 0$, $y_n = 1$ і нерівність $-y_1 + y_n \geqslant 1$ справджується як рівність. Зрештою, вартість розглядуваного розв'язку дорівнює

$$\sum_{(i,j)\in E} b_{ij}z_{ij} = \sum_{(i,j)\in E:\, i\in W,\, j\in\overline{W}} b_{ij} = C(W,\overline{W}).$$

□

Із леми 4.6 випливає, що величина будь-якого допустимого потоку не перевищує величини мінімального перерізу. Справді, за теоремою 1.4 вартість кожного допустимого розв'язку задачі (4.10) не може бути більшою від вартості будь-якого допустимого

розв'язку двоїстої задачі (4.11) і, зокрема, $C(W, \overline{W})$ — вартість розв'язку (4.12).

Наведемо достатні умови оптимальності для допустимого потоку.

Лема 4.7. *Нехай для допустимого потоку x знайдеться переріз $(W, \overline{W})$ такий, що всі його прямі дуги (i, j) насичені: $x_{ij} = b_{ij}$, а зворотні — порожні: $x_{ij} = 0$. Тоді x — максимальний потік величини $v = C(W, \overline{W})$.*

Доведення. Складемо рівності з обмеження задачі (4.10), які мають номери $i \in W$. Для будь-якої дуги $(i, j) \in E$, для якої $i, j \in W$, змінна x_{ij} входить у ліву частину отриманої рівності двічі: зі знаком плюс (коли дуга виходить з вершини i) і зі знаком мінус (коли дуга входить у вершину j). Тому такі змінні скоротяться. Звідси отримуємо рівність

$$\sum_{(i,j)\in E:\, i\in W, j\in\overline{W}} x_{ij} - \sum_{(i,j)\in E:\, i\in\overline{W}, j\in W} x_{ij} - v = 0$$

або, врахувавши умови леми,

$$v = \sum_{(i,j)\in E:\, i\in W, j\in\overline{W}} b_{ij} = C(W, \overline{W}).$$

За лемою 4.6 знайдемо допустимий розв'язок двоїстої задачі (4.11) вартості $C(W, \overline{W})$. За теоремою 1.4, x — максимальний потік. □

Залишилося навести метод побудови допустимого потоку, який задовольняє достатню умову оптимальності. Це можна зробити за допомогою алгоритму позначок Форда-Фалкерсона. Головним етапом цього алгоритму є процедура побудови збільшуючого шляху для довільного допустимого потоку.

Означення 4.9. *Нехай у мережі задані потік x і шлях* П*, який веде з вершини 1 до вершини n, в якому ігноруються напрями дуг. Шлях* П *називається збільшуючим, якщо він має такі властивостями:*

(i) *для кожної дуги* $(i,j) \in E$*, яка проходить шлях у прямому напрямі (яка називається прямою дугою),* $x_{ij} < b_{ij}$*. Остання нерівність означає, що прямі дуги шляху* Π *ненасичені;*

(ii) *для кожної дуги* $(i,j) \in E$*, яка проходять шляхом у зворотному напрямі (яка називається зворотною дугою),* $x_{ij} > 0$*. Отож, зворотні дуги шляху* Π *не повинні бути порожними.*

Позначимо через $E_1(\Pi)$ — множину всіх прямих дуг, а через $E_2(\Pi)$ — множину всіх зворотних дуг збільшуючого шляху Π. Визначимо величину

$$\delta = \min\{b_{ij} - x_{ij}, (i,j) \in E_1(\Pi);\, x_{ij}, (i,j) \in E_2(\Pi)\}.$$

Неважко бачити, що $\delta > 0$. Збільшимо на δ потік x_{ij}, який проходить по кожній прямій дузі (i,j) шляху Π, і зменшимо на цю ж величину потік x_{ij}, який проходить по кожній зворотній дузі. Покажемо, що умова збереження потоку в усіх проміжних вершинах шляху Π буде виконуватися. Для цього візьмемо три послідовні вершини i, j, k шляху Π. Чотири можливі випадки орієнтації дуг зображено на рис. 4.4.13.

Для кожного випадку додаткова кількість продукту, який входить до вершини j, дорівнює такій самій кількості продукту, що з неї виходить.

На рис. 4.4.14 зображено мережу з потоком $(3,1,2,1,1,2,2)$ величини 4.

Шлях $\Pi : 1 \to 3 \to 2 \to 4 \to 5$ є збільшуючим. Він містить зворотну дугу $(2,3)$ і $\delta = \min\{3,2,2,3\} = 2$. Після зміни потоку уздовж шляху Π отримаємо потік $(3,3,0,3,1,2,4)$ (рис. 4.4.15).

Опишемо процедуру побудови збільшуючого шляху. Вона використовує поняття позначки, яка приписується вершині мережі з деякої її сусідньої вершини. Така процедура буде полягати у просуванні позначок з вершини 1 до того моменту, коли або буде позначена вершина n, або процес зупиниться.

Нехай вершина j позначається з вершини i. Позначка вершини j складається з двох компонент: $L(j) = (L_1(j), L_2(j))$, які визначають так: якщо $(i,j) \in E$ і $x_{ij} < b_{ij}$, то $L_1(j) = i,\ L_2(j) =$

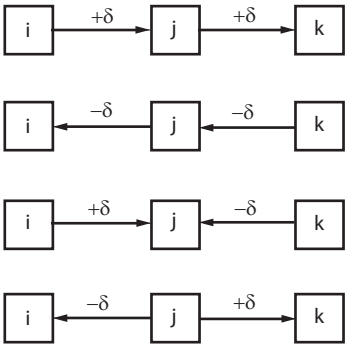


Рис. 4.4.13.

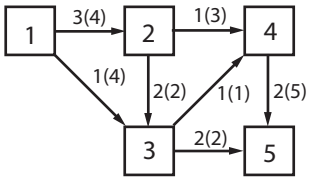


Рис. 4.4.14.

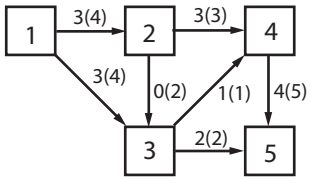


Рис. 4.4.15.

$\min[b_{ij} - x_{ij}, L_2(i)]$; якщо $(j,i) \in E$ і $x_{ji} > 0$, то $L_1(j) = -i$, $L_2(j) = \min[x_{ji}, L_2(i)]$. Отже, перша компонента позначки $L_1(j)$ містить інформацію, звідки позначається вершина j. Знак мінус в $L_1(j)$ означає, що позначку отримали вздовж зворотної дуги, тобто дуги, яка веде з вершини j до вершини i. Друга компонента $L_2(j)$ визначає величину додаткового потоку, який можна провести з вершини 1 до вершини j.

Процес поширення позначок з цієї вершини i називається *переглядом* вершини i. Будемо зберігати список S позначених, але не переглянутих вершин. Спочатку список S порожній. Процедура поширення позначок починає роботу з огляду вершини 1 і включення в кінець списку S всіх вершин, які позначені з початкової вершини 1. Потім процесс повторюється: вибирають і проглядають першу вершину i зі списку S, вершини, які позначені з i, додають у кінець списку S, а вершину i з S видаляють. Цей процес зупиняється у двох випадках: або кінцева вершина n отримує позначку, тоді можна відновити збільшуючий шлях зворотним переглядом з n з використанням компоненти L_1, або список S став порожнім.

Розглянемо **алгоритм позначок Форда-Фалкерсона.** Нехай x^1 — деякий початковий допустимий потік. Описаною вище процедурою будують збільшуючий шлях, за допомогою якого потік по мережі збільшується. Для отриманого потоку x^2 знову будується збільшуючий шлях до доти, доки у процедурі розстановки позначок список S не стане порожнім.

Теорема 4.1. *Якщо алгоритм Форда-Фалкерсона зупиняється на кроці k, то поточний потік x^k — максимальний.*

Доведення. Покажемо, що для потоку x^k справджується умова оптимальності. Визначимо W як множину всіх вершин, позначених на k-му кроці алгоритму, а $\overline{W} = V \setminus W$. Тоді для прямих дуг (i,j) перерізу $(W, \overline{W})$ виконується рівність $x^k_{ij} = b_{ij}$, оскільки в іншому випадку при перегляді вершини i позначку можна було поширити на вершину $j \in \overline{W}$, що суперечить визначенню множини W. Аналогічно для зворотних дуг перерізу $(W, \overline{W})$ виконується рівність $x^k_{ij} = 0$. Отже, умови леми 4.7 справджуються

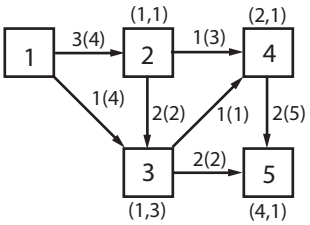


Рис. 4.4.16.

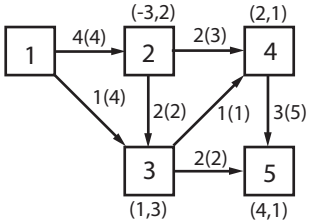


Рис. 4.4.17.

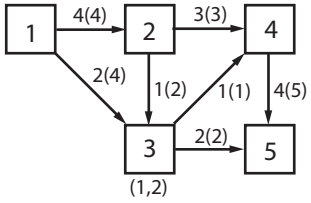


Рис. 4.4.18.

і потік x^k — максимальний. □

Зауважимо таке: якщо пропускні здатності дуг c_{ij}, $(i,j) \in E$, — раціональні числа, то алгоритм Форда-Фалкерсона зупиняється після скінченної кількості кроків.

Приклад 4.4. Застосуємо алгоритм Форда-Фалкерсона до мережі, що на рис. 4.4.14, з початковим потоком $x^1 = (3,1,2,1,1,2,2)$.

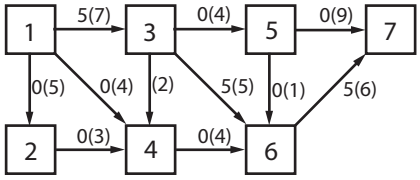


Рис. 4.4.19.

Позначку вершини будемо зазначати зверху або знизу від неї. Результати трьох кроків алгоритму зображено на рис. 4.4.16-4.4.18.

Для перших двох кроків отримані збільшуючі шляхи $1 \to 2 \to 4 \to 5$ і $1 \to 3 \to 2 \to 4 \to 5$ з $\delta = 1$, а на третьому кроці позначені лише вершини 1, 3 і 2. Вони і становитимуть множину W. Відповідний потік $x^3 = (4, 2, 1, 3, 1, 2, 4)$ величини $C(W, \overline{W}) = 6$ максимальний. Зауважимо, що переріз $(W, \overline{W})$ має лише прямі дуги $(2, 4)$, $(3, 4)$ і $(3, 5)$, які є насиченими. ▲

Задачі

4.16. Для мережі, зображеної на рис. 4.4.18, знайти всі перерізи пропускні здатності яких становить 7.

4.17. Нехай у потокової мережі з раціональними пропускними здатностями задано потік x^1 з раціональними компонентами. Довести, що алгоритм Форда - Фалкерсона, починаючи з потоку x^1, збігається за скінченну кількість етапів.

4.18. У гірській лісопарковій зоні розташовано сім майданчиків для відпочинку, з'єднаних стежками (рис. 4.4.19). На кожній дузі (i, j) в дужках зазначено максимальну (за один день), кількість туристичних груп, які можна провести від майданчика i до майданчика j по з'єднуючих їх стежках. Знайти максимальну (за один день) кількість груп, які можна провести від майданчика 1 до майданчика 7, використовуючи початковий потік x^1, в якому тільки такі компоненти додатні: $x^1_{13} = x^1_{36} = x^1_{57} = 5$. Які стежки можна закрити для використання без шкоди для туристичного бізнесу?

Список літератури

1. Бартіш М.Я. Дослідження операцій. Частина 1. Лінійні моделі: підручник/ М.Я.Бартіш,І.М.Дудзяний. - Львів: Видавничий центр ЛНУ імені Івана Франка, 2007.

2. Бартіш М.Я. Дослідження операцій. Частина 2. Алгоритми оптимізації на графах: підручник/ М.Я.Бартіш,І.М.Дудзяний. - Львів: Видавничий центр ЛНУ імені Івана Франка, 2007.

3. Бартіш М.Я. Дослідження операцій. Частина 3. Ухвалення рішень і теорія ігор: підручник/ М.Я.Бартіш,І.М.Дудзяний. - Львів: Видавничий центр ЛНУ імені Івана Франка, 2009.

4. Васин А.А. Исследование операций: учеб. пособ./ А.А.Васин,П.С.Краснощеков,В.В.Морозов. - М.:Академия, 2008.

5. Зайченко Ю.П. Дослідження операцій: підручник/ Ю.П.Зайченко. - К.: ЗАТ "ВІПОЛ 2000.

6. Крамер Н.Ш. Исследование операций в экономике: учебное пособие/ Н.Ш.Крамер,Б.А.Путко,И.М.Тришин. - М.: ЮНИТИ,2004.

7. Таха Х. Введение в исследование операций/Х.Таха; 7-е издание. - М.:Вильямс, 2005.

8. Шишкин Е.В. Математические методы и модели в управлении: учеб. пособ./ Е.В.Шишкин, А.Г.Чхартишвили. - М.:Дело, 2004.

9. Frederick S.Hilluer Introduction to operations research/Frederick S.Hilluer,Gerald J.Liberman. - McGraw-Hill, 2001

10. Kasana H.S. Introductory Operations Research. Theory and Applications/H.S.Kasana, K.D.Kumar. - Springer, 2004