

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**Київський національний університет
будівництва і архітектури**

**Кафедра інформаційних технологій проектування
та прикладної математики**

О.О. Терентьєв
доктор технічних наук, професор

ДОСЛІДЖЕННЯ ОПЕРАЦІЙ

КОНСПЕКТ ЛЕКЦІЙ

для здобувачів освітнього рівня «бакалавр»
спеціальності 122. «Комп'ютерні науки»,
123. «Комп'ютерна інженерія», 125. «Кібербезпека»
126. «Інформаційні системи і технології»,
015.10. «Професійна освіта. Комп'ютерні технології»

Київ, 2019 р.

Терентьєв О.О. Дослідження операцій: Конспект лекцій. – Київ: КНУБА, 2019. – 50 с.

Конспект лекцій розроблений на кафедрі інформаційних технологій проектування та прикладної математики Київського національного університету будівництва і архітектури для студентів денної форми навчання спеціальностей: 122. «Комп'ютерні науки», 123. Комп'ютерна інженерія, 125. «Кібербезпека», 126. «Інформаційні системи і технології», 015.10. «Професійна освіта. Комп'ютерні технології».

Затверджено на засіданні кафедри інформаційних технологій проектування та прикладної математики, протокол № 1 від 09.09.2019 р.

Рецензент: завідувач кафедри інформаційних технологій, доктор технічних наук, професор Цюцюра С.В.

ЗМІСТ

ВСТУП

Розділ 1. ГОЛОВНІ ПОНЯТТЯ ОПТИМІЗАЦІЇ

1.1. ЗАДАЧІ ОПТИМІЗАЦІЇ

1.2. ОПТИМІЗАЦІЙНІ ДОСЛІДЖЕННЯ

1.3. МЕТОДИ ПОБУДОВИ МАТЕМАТИЧНИХ МОДЕЛЕЙ

КОНТРОЛЬНІ ЗАПИТАННЯ

Розділ 2. ЛІНІЙНЕ ПРОГРАМУВАННЯ

2.1. ПОСТАНОВКА ЗАДАЧІ

2.2. КАНОНІЧНА ФОРМА ЗАДАЧ ЛІНІЙНОГО ПРОГРАМУВАННЯ

2.3. СИМПЛЕКС-МЕТОД

2.4. ДВОЇСТІСТЬ У ЛІНІЙНОМУ ПРОГРАМУВАННІ

2.5. ДВОЇСТИЙ СИМПЛЕКС-МЕТОД

2.6. ЛІНІЙНЕ ЦІЛОЧИСЛОВЕ ПРОГРАМУВАННЯ

КОНТРОЛЬНІ ЗАПИТАННЯ

Розділ 3. ОПТИМІЗАЦІЯ НА МЕРЕЖАХ

3.1. ЗАГАЛЬНІ ПОЛОЖЕННЯ

3.2. ЗАДАЧА ПРО НАЙКОРОТШИЙ ЛАНЦЮГ. АЛГОРИТМ
ДЕЙКСТРИ

3.3. ЗАДАЧА ПРО БАГАТОПОЛЮСНУ МЕРЕЖУ. АЛГОРИТМ
ФЛОЙДА

3.4. ЗАДАЧА ПРО ОПТИМАЛЬНЕ ДЕРЕВО-КІСТЯК

3.5. ЗАДАЧА ПРО ПРИЗНАЧЕННЯ. УГОРСЬКИЙ АЛГОРИТМ

3.6. ЗАДАЧА ПРО МАКСИМАЛЬНИЙ ПОТІК

КОНТРОЛЬНІ ЗАПИТАННЯ

ЛІТЕРАТУРА

ВСТУП

У наш час методи оптимізації застосовуються у багатьох областях практичної і наукової діяльності: проектування й аналіз технічних систем, планування виробництва, управління запасами і трудовими ресурсами та ін. Незважаючи на певну традиційність викладання питань, пов'язаних з оптимізацією, постійно виникає необхідність у подальшому розвитку цього напрямку, що обумовлено появою нових практичних задач, застосуванням потужних комп'ютерних пакетів: Mathlab, Mathcad, Derive, Mathematica та ін. Існує значна кількість підручників і посібників з оптимізації. На жаль, вони малодоступні для більшості студентів. Це, а також бажання зробити курс якомога більш зрозумілим, забезпечити у стислому вигляді викладання теоретичного і практичного матеріалу дисципліни, забезпечити більш ефективне проведення навчальних занять, привело до ідеї створення даного посібника.

Навчальний посібник підготовлено по матеріалах лекційних курсів, які автор протягом багатьох років викладає в Київському національному університеті будівництва і архітектури на факультеті автоматизації й інформаційних технологій. Також врахований досвід проведення практичних, лабораторних й індивідуальних занять, виконання курсових робіт.

Посібник складається з шести глав. Перші чотири глави охоплюють розгляд теоретичних і практичних питань таких розділів оптимізації: основи побудови математичних моделей, лінійне програмування, оптимізація на графах, нелінійне програмування. У цих розділах розглядаються оптимізаційні алгоритми, приклади розв'язання задач за цими алгоритмами, наводяться контрольні запитання з теоретичного матеріалу. П'ята глава містить задачі за темами, які розглянуті у попередніх главах. Ці задачі можуть використовуватися на практичних заняттях і як індивідуальні завдання. У шостій

главі наведено задачі, які можуть використовуватися для виконання курсових робіт. Кожна задача подана у двох варіантах. В додатку детально перелічено вимоги до курсової роботи.

У викладанні матеріалу автор був обмежений кількістю лекційних годин і обмеженнями на обсяг навчального посібника. Тому значна частина сучасних алгоритмів оптимізації залишилась поза розглядом. Крім того, розгляд прикладів, що ілюструють роботу алгоритмів, виконується без пояснень. В кінці книги наведено список рекомендованої літератури, де всебічно розглядаються питання, що пов'язані з оптимізацією.

Посібник орієнтований в першу чергу на студентів технічних вузів, може бути корисним для студентів економічних спеціальностей, для інженерів, аспірантів, наукових співробітників.

Розділ 1. ГОЛОВНІ ПОНЯТТЯ ОПТИМІЗАЦІЇ

1.1. Задачі оптимізації

Термін *оптимізація* означає процес або послідовність операцій, які дозволяють отримати краще *розв'язання*.

Постановка задачі оптимізації передбачає визначення критерію оптимальності, параметрів оптимізації і обмежень. **Критерій оптимальності** визначає ознаки, на основі яких проводиться порівняльна оцінка допустимих розв'язків і вибір оптимального розв'язку. **Параметри оптимізації** являють собою незалежні змінні параметри, які повністю і однозначно визначають задачу, що розв'язується. **Обмеження** визначають залежності між параметрами, які повинні враховуватися при пошуку розв'язання. Критерієм оптимальності може бути вимога досягнення найбільшого або найменшого значення однією або декількома функціями параметрів оптимізації, які відображають кількісну міру досягнення мети оптимізації об'єкта, що розглядається. Кожна з таких функцій називається **цільовою**. Якщо цільова функція (ЦФ), єдина то задачу оптимізації називають **задачею математичного програмування**, а в противному випадку – **задачею багатокритеріальної (векторної) оптимізації**.

Задачі математичного програмування залежно від цільової функції і обмежень поділяють на такі класи:

- лінійне програмування – цільова функція і обмеження лінійні;
- нелінійне програмування – цільова функція або обмеження нелінійні;
- цілочисельне програмування – змінні можуть набувати тільки цілочислових значень;
- дискретне програмування – змінні можуть набувати тільки визначених дискретних значень із заданої множини;

- булеве програмування – змінні можуть набувати тільки два значення 0 або 1;
- параметричне програмування – коефіцієнти залежать від певного параметра;
- стохастичне програмування – цільова функція і обмеження вміщують випадкові величини, які отримують певні значення із заданою імовірністю;
- геометричне програмування – коефіцієнти цільової функції являють собою поліноміальні функції;
- динамічне програмування – багатокрокові процеси пошуку оптимального розв’язку;
- сепарабельне програмування – цільова функція являє собою суму функцій, різних для кожної змінної, обмеження можуть бути лінійні і нелінійні.

Математична постановка задачі оптимізації:

$$\begin{aligned}
 f(x_j) &\rightarrow \text{opt (min, max);} \\
 g_i(x_j) &\leq b_i; \\
 d_j &\leq x_j \leq D_j; \\
 i &= \overline{1, m}; \quad j = \overline{1, n}.
 \end{aligned}
 \tag{1.1}$$

Тут $f(x_j)$ – цільова функція; $g_i(x_j)$ – обмеження, що встановлюють залежності між змінними; x_j – змінні, які можуть набувати значень у певному діапазоні d_j і D_j .

1.2. Оптимізаційні дослідження

Математичне програмування складається з таких етапів:

1. Вивчення системи; визначення мети, яка повинна бути досягнута внаслідок розв’язання задачі і умов, які визначають функціонування даного об’єкта або процесу; опис вхідної та вихідної інформації; визначення можливих підходів до розв’язання задачі.

2. Формалізація задачі; побудова математичної моделі (визначення критеріїв оптимальності, параметрів, обмежень).
3. Аналіз виду і структури математичної моделі; вибір методу розв'язання задачі.
4. Розв'язання задачі.
5. Аналіз результатів розв'язання.
6. Прийняття оптимального розв'язку.

1.3. Методи побудови математичних моделей

Математична модель – система математичних співвідношень, яка описує об'єкт, що вивчається.

Для побудови математичних моделей вивчаються і аналізуються причино-наслідкові зв'язки, проводяться експерименти, використовуються аналогії. Побудова математичної моделі передбачає виконання наступних етапів:

- визначення змінних задачі;
- визначення цільової функції;
- визначення обмежень на параметри.

Приклад. Скласти математичну модель задачі. При забудові мікрорайону використано проекти будинків трьох типів. Кількість квартир у будівлі і вартість одного будинку для кожного з проектів задано в табл. 1.1.

Таблиця 1.1

Квартира	Тип будинку		
	1	2	3
Однокімнатна, шт.	50	50	40
Двокімнатна, шт.	80	40	60
Трикімнатна, шт.	20	30	50
Вартість будівлі, тис.грн	850	800	900

Скласти план забудівлі мікрорайону, що мінімізує витрати на будівництво, якщо потреба в одно-, дво- і трикімнатних квартирах відповідно 500, 700, 600 шт.

Визначимо змінні:

x_j – кількість будівель j -го типу, $j=1,2,3$.

Визначимо обмеження:

b_i – кількість необхідних квартир i -го типу, $i=1,2,3$.

Визначимо цільову функцію:

c_j – вартість однієї будівлі j -го типу, $j=1,2,3$.

Визначимо норми використання будівель:

a_{ij} – норма використання будівель j -го типу для отримання заданої кількості квартир i -го типу.

Одна будівля 1-го типу дозволяє забезпечити потребу в 50-ти однокімнатних квартирах, а x_1 будівель забезпечать потребу в $50x_1$ однокімнатних квартирах. З урахуванням того, що для інших будівель залежності аналогічні, а загальна потреба в однокімнатних квартирах повинна бути не менше 500 шт., по однокімнатних квартирах обмеження матиме вигляд:

$$50x_1 + 50x_2 + 40x_3 \geq 500.$$

Аналогічно можна скласти обмеження для інших квартир і записати залежність для цільової функції. Тоді математична модель задачі матиме вигляд:

$$850x_1 + 700x_2 + 400x_3 \rightarrow \min$$

$$50x_1 + 50x_2 + 40x_3 \geq 500$$

$$80x_1 + 40x_2 + 60x_3 \geq 700$$

$$20x_1 + 30x_2 + 50x_3 \geq 600.$$

Змінні повинні бути невід'ємні: $x_j \geq 0$, $j=1,2,3$.

Контрольні запитання

1. Яка задача називається оптимізаційною?
2. Сформулювати основну задачу оптимізації.
3. Навести класифікацію оптимізаційних задач.
4. Які головні етапи оптимізаційних досліджень?

5. Що таке параметри задачі?
6. Що таке математична модель задачі?
7. Що дає використання математичних моделей?
8. Яка послідовність формування математичної моделі?
9. Які існують типи моделей і як вибрати тип моделей при розв'язуванні прикладної задачі?
10. Навести приклади задач, в яких застосовується оптимізація.
11. Чи існує прямий зв'язок між складністю моделі і її корисністю?
12. Чи завжди оптимізаційні моделі забезпечують найкраще розв'язання практичної задачі?
13. Чи обов'язково модель повинна вміщувати обмеження?
14. Охарактеризувати сучасні оптимізаційні пакети програм.

Розділ 2. ЛІНІЙНЕ ПРОГРАМУВАННЯ

2.1. Постановка задачі

Лінійне програмування являє собою найбільш часто використовуваний розділ оптимізації.

Задача лінійного програмування має такий вигляд:

$$\sum_{j=1}^n c_j x_j \rightarrow \max (\min); \quad (2.1)$$

$$\sum_{j=1}^n a_{ij} x_j = b_i, \quad i = \overline{1, r}; \quad (2.2)$$

$$\sum_{j=1}^n a_{ij} x_j \leq b_i, \quad i = \overline{r+1, k}; \quad (2.3)$$

$$\sum_{j=1}^n a_{ij} x_j \geq b_i, \quad i = \overline{k+1, m}; \quad (2.4)$$

$$x_j \geq 0, \quad j = \overline{1, n}, \quad (2.5)$$

де x_j – змінні, які необхідно визначити і які впливають на значення ЦФ; b_i – i -й ресурс; a_{ij} – норма витрат ресурсу i на j -й виріб.

Задача містить обмеження на невід'ємність змінних, наявність яких визначається роботою симплекс-методу. Якщо при фізичній постановці задачі яка-небудь змінна, наприклад x_n є необмеженою по знаку, то її можна представити у вигляді $x_n = x_n^{(1)} - x_n^{(2)}$, де $x_n^{(1)}, x_n^{(2)} \geq 0$. Вважатимемо всі числа $b_i \geq 0$, $i = 1, \dots, m$. Цього можна добитися множенням обмеження, де $b_i \leq 0$, на "–1".

Задачу лінійного програмування (ЗЛП) можна розв'язувати аналітичними і графічними методами. До аналітичних методів відносять метод Гаусса-Жордана і симплекс-метод. Графічне розв'язання ЗЛП достатньо наочне,

але обмежене кількістю змінних. Спочатку ЗЛП зводять до канонічної форми, а потім розв'язують її.

2.2. Канонічна форма задач лінійного програмування

Канонічна форма задач лінійного програмування має такий вигляд:

$$\begin{aligned} \sum_{j=1}^n c_j x_j &\rightarrow \max (\min) ; \\ \sum_{j=1}^n a_{ij} x_j &= b_i, \quad i = \overline{1, m}; \\ x_j &\geq 0, \quad j = \overline{1, n}. \end{aligned}$$

Задача лінійного програмування зводиться до канонічної форми шляхом перетворення всіх обмежень, які мають вигляд нерівностей, в рівності. Це досягається за такими правилами (для задачі максимізації):

- для кожного обмеження вигляду (2.3) додається нова базисна змінна;
- для кожного обмеження вигляду (2.2) додається нова базисна змінна і ця змінна вводиться у ЦФ з коефіцієнтом "-M" (M-задача);
- для кожного обмеження вигляду (2.4) додаються дві змінні $x_j - x_i$ (x_i – для того щоб отримати рівність, а x_j – нова базисна змінна, яка вводиться у ЦФ з коефіцієнтом "-M" (M-задача).

Таким чином, для зведення ЗЛП до канонічного вигляду необхідно ввести в обмеження m базисних змінних. Задачу мінімізації можна або звести попередньо до задачі максимізації (шляхом множення ЦФ на "-1"), або змінні вводяться у ЦФ з коефіцієнтом "+M".

Приклад 1. Звести до канонічного вигляду таку систему лінійних рівнянь:

$$\begin{aligned}4x_1 + 5x_2 &\rightarrow \max \\2x_1 - x_2 &\leq 2 \\3x_1 + 4x_2 &\geq 5 \\x_1 + x_2 &= 3 \\x_1, x_2 &\geq 0.\end{aligned}$$

Задача лінійного програмування має канонічний вигляд:

$$\begin{aligned}4x_1 + 5x_2 - Mx_5 - Mx_6 &\rightarrow \max \\2x_1 - x_2 + x_3 &= 2 \\3x_1 + 4x_2 - x_4 + x_5 &= 5 \\x_1 + x_2 + x_6 &= 3 \\x_i &\geq 0, \quad i = \overline{1,6}.\end{aligned}$$

Приклад 2. Звести до канонічного вигляду таку систему лінійних рівнянь:

$$\begin{aligned}4x_1 + 5x_2 &\rightarrow \min \\2x_1 - x_2 &\leq 2 \\3x_1 + 4x_2 &\geq 5 \\x_1 + x_2 &= 3 \\x_1, x_2 &\geq 0.\end{aligned}$$

Задача лінійного програмування має канонічний вигляд:

$$\begin{aligned}4x_1 + 5x_2 + Mx_5 + Mx_6 &\rightarrow \min \\2x_1 - x_2 + x_3 &= 2 \\3x_1 + 4x_2 - x_4 + x_5 &= 5 \\x_1 + x_2 + x_6 &= 3 \\x_i &\geq 0, \quad i = \overline{1,6}.\end{aligned}$$

2.4. Симплекс-метод

Постановка задачі. Головна ідея симплекс-методу полягає в цілеспрямованому послідовному знаходженні екстремальних точок за допомогою допустимих базисних розв'язків системи обмежень. Змінні $x_1, x_2, \dots, x_m$ називаються **базисними**, якщо вони входять тільки в одне з рівнянь системи обмежень з коефіцієнтом 1, а в усі інші рівняння з коефіцієнтами 0. Всі інші $n-m$ змінних називаються **небазисними (вільними)**. **Базисним розв'язком** ЗЛП називається розв'язання:

$$x_i = b_i, i = 1, \dots, m; \quad x_{m+1} = x_{m+2} = \dots = x_n = 0.$$

Базисний розв'язок називається **допустимим**, якщо $x_i \geq 0$, $i = 1, \dots, m$. Базисний розв'язок називається **невиродженим**, якщо $x_i > 0$, $i = 1, \dots, m$.

Процес розв'язування починається з деякої початкової екстремальної точки (допустимий базисний розв'язок) і продовжується доти, поки не буде знайдена точка, яка відповідає оптимальному розв'язку. Як нову екстремальну точку обирають суміжну з попередньою екстремальну точку, в якій значення ЦФ щонайменше не погіршується. Максимальна кількість ітерацій не перевищує C_n^m (сполучення із n по m). Суміжні екстремальні точки відрізняються тільки однією змінною. Кожна наступна екстремальна точка визначається заміною однієї з поточних базисних змінних новою з числа небазисних.

Симплекс-метод передбачає виконання таких операцій:

- визначення початкового допустимого базисного розв'язку, при цьому $(n-m)$ небазисних змінних прирівнюються нулю і з ЦФ виключаються базисні змінні;
- визначення з числа небазисних змінних змінної, яку необхідно включити в число базисних змінних, ця нова базисна змінна повинна покращити значення ЦФ;

- визначення з числа базисних змінних змінної, яка повинна бути виключена з базисного розв'язку, тобто замінена новою базисною змінною.

Виконання розрахунків пов'язане з використанням симплекс-таблиць (табл. 2.1).

Таблиця 2.1

Симплекс-таблиця

Базис	$C_{\text{баз}}$	БР	c_1	c_2	$\dots$	c_n	c_{n+1}	$\dots$	c_{n+m}
			A_1	A_2	$\dots$	A_n	A_{n+1}	$\dots$	A_{n+m}
A_{n+1}	c_{n+1}	b_1	a_{11}	a_{12}	$\dots$	a_{1n}	a_{1n+1}	$\dots$	a_{1n+m}
A_{n+2}	c_{n+2}	b_2	a_{21}	a_{22}	$\dots$	a_{2n}	a_{2n+1}	$\dots$	a_{2n+m}
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
A_{n+m}	c_{n+m}	b_m	a_{m1}	a_{m2}	$\dots$	a_{mn}	a_{mn+1}	$\dots$	a_{mn+m}
$\Delta_j = z_j - c_j$		z_0	$z_1 - c_1$	$z_2 - c_2$	$\dots$	$z_n - c_n$	$z_{n+1} - c_{n+1}$	$\dots$	$z_{n+m} - c_{n+m}$

БР – базисні розв'язки.

Значення z_j визначаються так:

$$z_j = \sum_{i=1}^m c_{i_B} a_{ij}, \quad j = \overline{1, m+n},$$

де c_{i_B} – коефіцієнти ЦФ при поточних базисних змінних; i_B – індекс базисної змінної, що розташована в i -му рядку симплекс-таблиці; a_{ij} – елементи стовпця коефіцієнтів при змінній x_j в системі рівнянь, що відповідає поточному базису.

Алгоритм розв'язання задач симплекс-методом містить такі етапи (для задачі мінімізації):

Крок 1. Знайти початковий базисний розв'язок:

- записати вихідну канонічну задачу (простий симплекс-метод або M -задача);
- визначити базисні змінні, які входять тільки в одне з рівнянь системи з коефіцієнтом 1, а в усі інші – з коефіцієнтом, що дорівнює 0;
- виділити вільні змінні (всі інші, крім базисних);

- знайти початковий базисний розв'язок, вважаючи, що вільні змінні дорівнюють 0.

Крок 2. Заповнити симплекс-таблицю.

Обчислити відносні оцінки і записати їх у таблицю.

$$\Delta_j = \sum_{i=1}^m c_{i_B} a_{ij} - c_j = z_j - c_j,$$

$$z_j = \sum_{i=1}^m c_{i_B} a_{ij}; \quad j = \overline{1, m+n}.$$

Для базисних змінних оцінки дорівнюють 0. Це можна використовувати як для перевірки правильності заповнення таблиці, так і для скорочення обчислень.

Крок 3. Проаналізувати відносні оцінки.

1. Якщо всі оцінки Δ_j недодатні, тобто

$$\Delta_j \leq 0, \quad j = \overline{1, m+n},$$

то розрахунок закінчений і необхідно знайти отриманий базисний розв'язок. Значення базисних змінних знаходяться в стовпці БР, а інші змінні вважаються рівними 0, як вільні.

Проаналізувати отриманий базисний розв'язок:

- якщо число нульових оцінок $\Delta_j = 0$ дорівнює числу базисних змінних, то задача має **єдиний розв'язок**;
- якщо число нульових оцінок $\Delta_j = 0$ перевищує число базисних змінних, то задача має **нескінченну безліч розв'язків**;
- якщо усі Δ_j – недодатні, але базисний розв'язок містить хоча б одну штучну змінну, яка не дорівнює 0, то **обмеження задачі несумісні**.

2. Якщо серед оцінок є додатні, то необхідно знайти серед них максимальну:

$$\Delta_r = \max \Delta_j,$$

і проаналізувати коефіцієнти стовпця таблиці, якому відповідає максимальна додатна оцінка (якщо таких оцінок декілька, то

прийнято вибирати оцінку з найменшим номером). Якщо цей стовпець містить хоча б один додатний коефіцієнт, то номер стовпця позначається через r , і змінна, яка йому відповідає, повинна бути введена в число базисних. Якщо серед коефіцієнтів цього стовпця немає жодного додатного коефіцієнта, то це означає, що множина допустимих розв'язків задачі є необмеженою, і **задача розв'язку не має**.

Стовпець, що відповідає обраній оцінці, називається **розв'язувальним**.

Крок 4. Поділити елементи стовпця базисних розв'язків на відповідні елементи розв'язувального стовпця і серед отриманих часток вибрати найменшу. Рядок, що відповідає обраному відношенню називається **розв'язувальним**.

Таким чином, нова змінна x_r вводиться на місце змінної x_{s_B} , що вилучається з числа базисних, номер змінної s_B , а також номер s відповідного рядка таблиці, визначаються з умови

$$\min_{1 \leq i \leq m} \left[\frac{x_{i_B}}{a_{ir}} \right] = \frac{x_{s_B}}{a_{sr}},$$

де x_{s_B} – значення координати поточного базисного розв'язку, що відповідає i -му рядку; a_{ir} – коефіцієнт при координаті x_r в i -му рядку. Якщо таких змінних виявиться більше однієї, то з базису виводиться та змінна, яка має більший номер. Розглядаються тільки невід'ємні відношення, тобто якщо коефіцієнт a_{ir} – від'ємний або дорівнює 0, то відношення не підраховується.

Елемент a_{sr} , розташований на перетинанні розв'язувального рядка і розв'язувального стовпця, називається **розв'язувальним** і виділяється.

З числа базисних виводиться змінна, що відповідає розв'язувальному рядку, а на її місце вводиться змінна, що відповідає розв'язувальному стовпцю.

Крок 5. Обчислити новий базисний розв'язок, здійснивши перерахування таблиці:

1. Замість координати x_{s_B} до складу базисних увести координату x_r , значення якої визначається за формулою

$$x_r = \frac{x_{s_B}}{a_{sr}},$$

і перерахувати s -й рядок, у якому відбулися зміни по базису:

$$a_{sj} = \frac{a_{sj}}{a_{sr}}, \quad j = \overline{1, m+n}.$$

Таким чином, кожен елемент розв'язувального рядка, поділяється на розв'язувальний елемент a_{sr} ;

2. Обчислити всі інші коефіцієнти:

$$a_{ij} = a_{ij} - \frac{a_{sj}}{a_{sr}} a_{ir},$$

$$i = 1, \dots, m; \quad i \neq s; \quad j = \overline{1, m+n}.$$

Новий базисний розв'язок визначити на основі поточного базисного розв'язку за формулами

$$x_{i_B} = x_{i_B} - a_{ir} x_r, \quad i_B \neq s_B.$$

Для спрощення обчислень за наведеними формулами використовується "правило прямокутника".

Нехай підраховується значення a_{ij} . Варто з'єднати елемент a_{ij} у попередній таблиці з розв'язувальним елементом a_{sr} . Отримано одну з діагоналей прямокутника. Другу діагональ утворить з'єднання елементів a_{ir} і a_{sj} . Далі з поточного значення a_{ij} віднімається добуток елементів a_{ir} і a_{sj} , поділений на розв'язувальний елемент a_{sr}

$$a_{ij} = a_{ij} - \frac{a_{sj} \cdot a_{ir}}{a_{sr}}$$

a_{sr}	a_{sj}
a_{ir}	a_{ij}

Перейти до кроку 3.

Приклад 1. Розв'язати симплекс-методом ЗЛП.

$$2x_1 - x_2 \rightarrow \max$$

$$-x_1 + x_2 \leq 3;$$

$$6x_1 + 7x_2 \leq 42$$

$$2x_1 - 3x_2 \leq 6$$

$$x_i \geq 0, i = 1, 2.$$

1. Зведемо до канонічного вигляду:

$$-x_1 + x_2 + x_3 = 3$$

$$6x_1 + 7x_2 + x_4 = 42$$

$$2x_1 - 3x_2 + x_5 = 6$$

$$x_i \geq 0, i = \overline{1, 5}.$$

де x_3, x_4, x_5 – додаткові змінні, додані в нерівність, щоб перетворити їх у рівність.

2. Результати розв'язування представлені в табл.2.2–2.4.

Таблиця 2.2

Базис	$C_{\text{баз}}$	БР	2	-1	0	0	0
			A_1	A_2	A_3	A_4	A_5
A_3	0	3	-1	1	1	0	0
A_4	0	42	6	7	0	1	0
A_5	0	6	2	-3	0	0	1
$\Delta_j = Z_j - C_j$		0	-2	1	0	0	0

Таблиця 2.3

Базис	$C_{\text{баз}}$	БР	2	-1	0	0	0
			A_1	A_2	A_3	A_4	A_5
A_3	0	6	0	-0,5	1	0	0,5
A_4	0	24	0	16	0	1	-3
A_1	-2	3	1	-1,5	0	0	0,5
$\Delta_j = Z_j - C_j$		6	0	-3	0	0	1

Таблиця 2.4

Базис	$C_{\text{баз}}$	БР	2	-1	0	0	0
			A_1	A_2	A_3	A_4	A_5
A_3	0	6,75	0	0	1	0,031	0,406
A_2	-1	1,5	0	1	0	0,063	-0,19
A_1	2	5,25	1	0	0	0,094	0,22
$\Delta_j = Z_j - C_j$		9	0	0	0	0,125	0,625

Згідно з алгоритмом розв'язування задачі знайдено:

$$x_1 = 5,25; x_2 = 1,5; x_3 = 6,75; x_4 = 0; x_5 = 0.$$

Значення ЦФ: $z = 9$.

Приклад 2. Розв'язати симплекс-методом M -задачу лінійного програмування.

$$\begin{aligned} 5x_1 + 2x_2 - x_3 &\rightarrow \max \\ -2x_1 - x_2 - x_3 &\geq -5 \\ 3x_1 + 2x_2 + x_3 &= 6 \\ 5x_1 + 3x_2 + 4x_3 &\geq 1 \\ x_i &\geq 0, i = \overline{1,3}. \end{aligned}$$

1. Зведемо до канонічного вигляду:

$$\begin{aligned} 5x_1 + 2x_2 - x_3 - Mx_6 - Mx_7 &\rightarrow \max \\ 2x_1 + x_2 + x_3 + x_4 &= 5 \\ 3x_1 + 2x_2 + x_3 + x_6 &= 6 \\ 5x_1 + 3x_2 + 4x_3 - x_5 + x_7 &= 1 \\ x_i &\geq 0, \quad i = \overline{1,3}, \end{aligned}$$

де x_4, x_5, x_6, x_7 – додаткові змінні, додані в нерівності, щоб перетворити їх у рівності: x_4, x_6, x_7 – початкові, допустимі базисні розв'язки.

3. Результати розв'язування представлені в табл.2.5–2.7.

Таблиця 2.5

Базис	$C_{\text{баз}}$	БР	5	2	-1	0	0	-M	-M
			A_1	A_2	A_3	A_4	A_5	A_6	A_7
A_4	0	5	2	1	1	1	0	0	0
A_6	-M	6	3	2	1	0	0	1	0
A_7	-M	1	5	3	4	0	-1	0	1
$\Delta_j = Z_j - C_j$		-7M	-5 -8M	-2 -5M	1-5M	0	M	0	0

Таблиця 2.6

Базис	$C_{\text{баз}}$	БР	5	2	-1	0	0	-M	-M
			A_1	A_2	A_3	A_4	A_5	A_6	A_7
A_4	0	23/5	0	-1/5	-3/5	1	2/5	0	-2/5
A_6	-M	27/5	0	1/5	-7/5	0	3/5	1	-3/5
A_1	5	1/5	1	3/5	4/5	0	-1/5	1	1/5
$\Delta_j = Z_j - C_j$		27M/5 +1	0	-M/5+1	7M/5 +5	0	1 -3M/5	0	8M/5 +1

Таблиця 2.7

Базис	$C_{\text{баз}}$	БР	5	2	-1	0	0	-M	-M
			A_1	A_2	A_3	A_4	A_5	A_6	A_7
A_4	0	1	0	-1/3	1/3	1	0	-2/3	0
A_5	0	9	0	1/3	-7/3	0	1	5/3	-1
A_1	5	2	1	2/3	1/3	0	0	1/3	0
$\Delta_j = Z_j - C_j$		10	0	4/3	8/3	0	0	-5/3 +M	2+M

Згідно з алгоритмом розв'язування задачі знайдено:

$$x_1 = 2; x_2 = 0; x_3 = 0; x_4 = 1; x_5 = 9; x_6 = 0; x_7 = 0.$$

Значення ЦФ: $z = 10$.

2.5. Двоїстість у лінійному програмуванні

Кожній вихідній (прямій) ЗЛП відповідає *двоїста задача*. Розв'язання однієї задачі є одночасно розв'язанням і іншої. Поняття двоїстості використовується при дослідженні ЗЛП на чутливість, дозволяє дати інтерпретацію отриманим розв'язкам, розкрити нові закономірності і співвідношення, які властиві проблемі, що досліджується. Змінні і обмеження двоїстої задачі формуються шляхом симетричних структурних перетворень прямої задачі за такими правилами.

1. Кожному i -му обмеженню прямої задачі відповідає змінна u_i двоїстої задачі і, навпаки, кожному j -му обмеженню двоїстої задачі відповідає змінна x_j прямої задачі.

2. Матриця коефіцієнтів при двоїстих змінних в обмеженнях двоїстої задачі є транспонованою матрицею коефіцієнтів при змінних в обмеженнях вихідної задачі.

3. Вільні члени обмежень однієї із задач є коефіцієнтами при відповідних змінних в ЦФ іншої задачі; максимізація змінюється на мінімізацію і навпаки.

4. Якщо в прямій задачі обмеження мають знаки нерівностей $\leq$, то в двоїстій задачі вони будуть $\geq$.

Для прямої ЗЛП такого виду:

$$\begin{aligned} \sum_{j=1}^n c_j x_j &\rightarrow \max; \\ \sum_{j=1}^n a_{ij} x_j &\leq b_i, \quad i = \overline{1, m}; \\ x_j &\geq 0, \quad j = \overline{1, n}, \end{aligned}$$

існує двоїста ЗЛП (ДЗЛП):

$$\begin{aligned} \sum_{i=1}^m b_i y_i &\rightarrow \min; \\ \sum_{i=1}^m a_{ij} y_i &\geq b_j, \quad j = \overline{1, n}; \\ y_i &\geq 0, \quad i = \overline{1, m}. \end{aligned}$$

Приклад. Звести пряму ЗЛП до ДЗЛП:

$$\begin{aligned} x_1 - 4x_2 - 3x_3 &\rightarrow \min \\ 3x_1 + 4x_2 + x_3 &\leq 7 \\ x_1 + 2x_2 + x_3 &= 6 \\ x_3 &\geq 4 \\ x_j &\geq 0, \quad j = \overline{1, 3}. \end{aligned}$$

Виконаємо попередні перетворення задачі :

- задачу мінімізації перетворимо на задачу максимізації;
- знаки " $\geq$ " замінимо на знаки " $\leq$ ";
- знаки "=" замінимо на знаки " $\leq$ " і замість одного рівняння введемо двох.

$$\begin{aligned} -x_1 + 4x_2 + 3x_3 &\rightarrow \max \\ 3x_1 + 4x_2 + x_3 &\leq 7 \\ x_1 + 2x_2 + x_3 &\leq 6 \\ -x_1 - 2x_2 - x_3 &\leq -6 \\ -x_3 &\leq -4 \\ x_j &\geq 0, \quad j = \overline{1, 3}. \end{aligned}$$

Побудуємо двоїсту задачу до прямої за вищенаведеними правилами:

$$\begin{aligned}
& 7y_1 + 6y_2 - 6y_3 - 4y_4 \rightarrow \min \\
& 3y_1 + y_2 - y_3 \geq -1 \\
& 4y_1 + 2y_2 - 2y_3 \geq 4 \\
& y_1 + y_2 - y_3 - y_4 \geq 3 \\
& y_i \geq 0, \quad i = \overline{1, 4}.
\end{aligned}$$

2.6. Двоїстий симплекс-метод

В основі двоїстого симплекс-методу (ДСМ) лежить ідея побудови на кожній ітерації, за винятком останньої, розв'язку, який є недопустимим внаслідок невиконання умов невід'ємності змінних; при цьому відповідні розв'язки двоїстої задачі на кожній ітерації є допустимими. ДСМ – це застосування симплекс-методу до двоїстої задачі, доповнене побудовою на кожній ітерації вектора змінних, який є майже допустимим опорним розв'язком вихідної задачі. ДСМ використовується при розв'язуванні ЗЛП у таких випадках:

- початковий опорний допустимий базисний розв'язок не є очевидним (наприклад, метод штучних змінних);
- після отримання розв'язків необхідно додати до вихідних обмежень додаткові обмеження (наприклад, розв'язки задач лінійного цілочисельного програмування).

Базисом опорного плану $Y = (y_1, y_2, \dots, y_m)$, або **сполученим базисом**, називається система m незалежних векторів обмежень прямої задачі, для яких m обмежень двоїстої задачі виконуються як строгі рівності.

Вектор $X = (x_1, x_2, \dots, x_n)$ компоненти якого дорівнюють коефіцієнтам розкладу змінних x по векторах сполученого базису називається **псевдопланом вихідної задачі**. Псевдоплан X буде оптимальним розв'язком вихідної задачі, якщо серед його базисних компонентів немає від'ємних. План Y , який визначає псевдоплан X , є розв'язком двоїстої задачі.

Алгоритм двоїстого симплекс-методу.

Крок 1. Представити вихідну задачу лінійного програмування у канонічному вигляді.

Крок 2. Записати двоїсту задачу до заданої задачі.

Крок 3. Визначити базис опорного плану для двоїстої задачі.

Крок 4. Визначити коефіцієнти розкладу небазисних векторів по векторах сполученого базису.

Крок 5. Записати псевдоплан $X = (x_1, x_2, \dots, x_n)$.

Крок 6. Перевірити, чи існують серед компонентів псевдоплану $X = (x_1, x_2, \dots, x_n)$ від'ємні. Якщо їх не існує, то план є оптимальним, інакше переходимо до кроку 7.

Крок 7. Перевірити в рядках псевдоплану, які відповідають його від'ємним компонентам b_i , чи всі елементи $a_{ij} \geq 0$, для $j=1, \dots, n$. Якщо так, то задача не має розв'язку. В протилежному випадку переходимо до кроку 8.

Крок 8. Вибрати розв'язувальний рядок, тобто вектор, який буде виведено з базису. Якщо існує декілька $b_i \leq 0$, то обирається рядок з найменшим значенням b_i .

Крок 9. Вибрати розв'язувальний стовпець, тобто вектор який буде введено в базис. Номер цього стовпця визначається із співвідношення:

$$\min \left\{ -\frac{\Delta_j}{a_{ij}} \mid a_{ij} < 0 \right\}.$$

Крок 10. Виконуються симплексні перетворення і перехід до кроку 6.

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

$$2x_1 + x_2 \rightarrow \max$$

$$x_1 + x_2 \leq 6$$

$$2x_1 + 8x_2 \geq 16$$

$$15x_1 - 3x_2 \geq 15$$

$$x_i \geq 0, \quad i = 1, 2.$$

Запишемо задачу в канонічному вигляді:

$$2x_1 + x_2 \rightarrow \max$$

$$x_1 + x_2 + x_3 = 6$$

$$-2x_1 - 8x_2 + x_4 = -16$$

$$-15x_1 + 3x_2 + x_5 = -15$$

$$x_i \geq 0, \quad i = \overline{1, 5}.$$

Двоїста задача до заданої задачі:

$$6y_1 - 16y_2 - 15y_3 \rightarrow \min$$

$$y_1 - 2y_2 - 15y_3 \geq 2$$

$$y_1 - 8y_2 + 3y_3 \geq 1$$

$$y_j \geq 0, \quad j = \overline{1, 3}.$$

Псевдоплан має вигляд $X = (0, 0, 6, -16, -15)$.

Заповнимо симплекс-таблицю і виконаємо розв'язання згідно з алгоритмом. Результати розв'язання наведені в табл.2.8–2.10.

Таблиця 2.8

Базис	$C_{\text{баз}}$	БР	2	1	0	0	0
			A_1	A_2	A_3	A_4	A_5
A_3	0	6	1	1	1	0	0
A_4	0	-16	-2	-8	0	1	0
A_5	0	-15	-15	3	0	0	1
$\Delta_j = Z_j - C_j$		0	-2	-1	0	0	0

На першій ітерації задача розв'язується симплекс-методом. Розв'язувальний елемент дорівнює 1.

Таблиця 2.9

Базис	$C_{\text{баз}}$	БР	2	1	0	0	0
			A_1	A_2	A_3	A_4	A_5
A_1	2	6	1	1	1	0	0
A_4	0	-4	0	-6	2	1	0
A_5	0	75	0	18	16	0	1
$\Delta_j = Z_j - C_j$		12	0	1	2	0	0

На другій ітерації задача розв'язується двоїстим симплекс-методом.

Таблиця 2.10

Базис	$C_{\text{баз}}$	БР	2	1	0	0	0
			A_1	A_2	A_3	A_4	A_5
A_1	2	32/6	1	0	4/3	1/6	0
A_2	1	2/3	1	1	-1/3	-1/6	0
A_5	0	63	0	0	22	3	1
$\Delta_j = Z_j - C_j$		34/3	0	0	7/3	1/6	0

Розв'язки: $X = (32/6, 2/3, 0, 0, 63)$, $Y = (7/3, 1/6, 0)$, $z = 34/3$.

2.7. Дослідження на чутливість

Аналіз чутливості виконується після отримання оптимального розв'язку задачі оптимізації. Мета аналізу чутливості – визначити, чи приведе зміна коефіцієнтів вихідної задачі до зміни поточного оптимального розв'язку, і якщо так, то як ефективно знайти новий оптимальний розв'язок, у тому випадку якщо він існує.

Виконуючи аналіз на чутливість при відомому оптимальному розв'язку, в першу чергу, визначають діапазони допустимих змін коефіцієнтів при змінних у ЦФ. При цьому

$$x_3 = 6,75 + 0,406\varepsilon_3 \geq 0;$$

$$x_2 = 1,5 - 0,19\varepsilon_3 \geq 0;$$

$$x_1 = 5,25 + 0,22\varepsilon_3 \geq 0.$$

Розв'язуючи систему рівнянь, отримаємо: $7,89 \geq \varepsilon_3 \geq -16,63$.

При зміні ε_3 в цьому діапазоні значення ЦФ буде змінюватися за правилом $z = 9 + 0,625\varepsilon_3$.

2.8. Лінійне цілочислове програмування

Задача цілочислового лінійного програмування (ЦЛП) описується моделлю такого вигляду:

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

$$\sum_{j=1}^n a_{ij} x_j \leq b_i, \quad i = \overline{1, m};$$

$$x_j \geq 0, \quad j = \overline{1, n};$$

$$x_j - \text{цілі числа}, \quad j = 1, \dots, n.$$

Округлення результатів розв'язання задачі може привести або до недопустимого розв'язку, або до результатів, які суттєво відрізняються від оптимальних.

Всі методи розв'язування задач ЦЛП поділяються на дві групи: **методи відтинання і комбінаторні методи**. В **методах відтинання** на кожній ітерації додається лінійне обмеження, яке задовольняє цілочислового розв'язання вихідної задачі і виключає поточне нецілочислове розв'язання попередньої ітерації. Збіжність досягається по кінцевій кількості ітерацій, але це число може бути дуже великим. Серед методів відтинання найбільш відомі **перший метод Гоморі** (для розв'язування повністю цілочислових задач) і **другий алгоритм Гоморі** (для розв'язування частково цілочислових задач). Всі **комбінаторні методи** базуються на

ідеї перебору всіх допустимих цілочислових розв'язків. Серед комбінаторних методів найбільш відомим є *метод гілок і границь*.

Перший алгоритм Гоморі передбачає побудову відтинання за такими правилами. Рівняння

$$\sum_{j=1}^n a_{ij}x_j = b_i ,$$

яке містить дробові значення базисних змінних $x_i = b_i$, перетворюється у додаткову нерівність:

$$\sum_{j=1}^n \alpha_{ij}x_j \geq \beta_i ,$$

де $\alpha_{ij} = a_{ij} - [a_{ij}]$; $\beta_{ij} = b_{ij} - [b_{ij}]$; $\alpha_{ij} \geq 0$; $\beta_{ij} \geq 0$.

Тут α_{ij} – дробова частина числа a_{ij} ; $[a_{ij}]$ – ціла частина числа a_{ij} ; β_{ij} – дробова частина числа b_{ij} ; $[b_{ij}]$ – ціла частина числа b_{ij} .

Перший алгоритм Гоморі.

Крок 1. Розв'язати ЗЛП без урахування вимог на цілочисловість змінних. Якщо розв'язання задачі цілочислове, то обчислення завершено. В протилежному випадку перейти до кроку 2.

Крок 2. Вибрати в останній симплекс-таблиці нецілу змінну x_i з максимальною дробовою частиною.

Крок 3. Записати використовуючи останню симплекс-таблицю рівняння відтинання в дробових частинах:

$$-\sum_{j=1}^n \alpha_{ij}x_j + x_i = -\beta_i.$$

Крок 4. Розширити останню симплекс-таблицю на один рядок і один стовпець, записати в цю таблицю додаткові обмеження.

Крок 5. Розв'язати псевдоплан двоїтим симплекс-методом. Перейти до кроку 1.

Приклад. Розв'язати цілочислову задачу лінійного програмування:

$$\begin{aligned} 3x_1 + 2x_2 &\rightarrow \max \\ x_1 + x_2 &= 11 \\ x_1 - x_2 &\leq 6 \\ x_1 &\geq 3 \\ x_j &\geq 0, \quad j=1, 2 \\ x_j &\text{ – цілі числа, } j = 1, 2. \end{aligned}$$

Зведемо задачу до канонічного вигляду:

$$\begin{aligned} 3x_1 + 2x_2 - Mx_3 - Mx_5 &\rightarrow \max \\ x_1 + x_2 + x_3 &= 11 \\ x_1 - x_2 + x_4 &= 6 \\ x_1 - x_6 + x_5 &= 3 \\ x_j &\geq 0, \quad j=1, \dots, 6 \\ x_j &\text{ – цілі числа, } j=1, \dots, 6. \end{aligned}$$

Результати розв'язання представлені в табл.2.14–2.19.

Таблиця 2.14

Базис	$C_{\text{баз}}$	БР	3	2	-M	0	-M	-M
			A_1	A_2	A_3	A_4	A_5	A_6
A_3	-M	11	1	1	1	0	0	0
A_4	0	6	1	-1	0	1	0	0
A_5	-M	3	1	0	0	0	1	-1
$\Delta_j = Z_j - C_j$		0 -14M	-3 -2M	-2 -M	0 -M	0 0M	0 -M	0 M

Таблиця 2.15

Базис	$C_{\text{баз}}$	БР	3	2	-M	0	-M	-M
			A_1	A_2	A_3	A_4	A_5	A_6
A_3	-M	8	0	1	1	0	-1	1
A_4	0	3	0	-1	0	1	-1	1
A_1	3	3	1	0	0	0	1	-1
$\Delta_j = Z_j - C_j$		9 -8M	3 0M	-2 -M	0 -M	0 0M	3 M	-3 -M

Таблиця 2.16

Базис	$C_{\text{баз}}$	БР	3	2	-M	0	-M	-M
			A_1	A_2	A_3	A_4	A_5	A_6
A_2	2	8	0	1	1	0	-1	1
A_4	0	11	0	0	1	1	-2	2
A_1	3	3	1	0	0	0	1	-1
$\Delta_j = Z_j - C_j$		25 0M	0 0M	0 0M	2 0M	0 0M	1 0M	-1 0M

Таблиця 2.17

Базис	$C_{\text{баз}}$	БР	3	2	-M	0	-M	-M
			A_1	A_2	A_3	A_4	A_5	A_6
A_2	2	5/2	0	1	1/2	-1/2	0	0
A_6	0	11/2	0	0	1/2	1/2	-1	1
A_1	3	17/2	1	0	1/2	1/2	0	0
$\Delta_j = Z_j - C_j$		61/2	0	0	5/2	1/2	0	0

Розв'язки нецілочислової задачі: $X = (17/2, 5/2, 0, 0, 0, 11/2)$, $z = 61/2$.

Відтинання для змінної x_1 на основі останньої симплекс-таблиці (для базису A_1):

$$-1/2x_3 - 1/2x_4 + x_7 = -1/2$$

Додамо це обмеження до системи обмежень вихідної задачі і вирішуватимемо задачу із застосуванням двоїстого симплекс-методу.

Розділ 3. ОПТИМІЗАЦІЯ НА МЕРЕЖАХ

3.1. Загальні положення

Мережі використовуються для розв'язування задач оптимізації в різних предметних областях. Ці задачі мають певну специфіку формалізації і розв'язання:

- значна частина задач цього типу може вирішуватися методами лінійного програмування, однак специфічні особливості цих задач дозволили розробити більш ефективні методи;
- ці задачі мають геометричне зображення у вигляді графа.

В цьому розділі розглядаються такі задачі:

- задача про найкоротший ланцюг;
- задача про багатополюсну мережу;
- задача про оптимальне дерево-кістяк;
- транспортна задача;
- задача про призначення;
- задача про максимальний потік.

3.2. Задача про найкоротший ланцюг.

Алгоритм Дейкстри

Постановка задачі. Необхідно визначити в мережі найкоротший ланцюг між заданими вхідним пунктом і пунктом призначення.

Алгоритм Дейкстри.

Алгоритм будується на положенні, що якщо найкоротший ланцюг із вузла i у вузол j проходить через вузол k , то найкоротший ланцюг із вузла i у вузол k є частиною початкового найкоротшого ланцюга, який закінчується у вузлі k .

У процесі виконання цього алгоритму використовується процедура приписування міток вузлам. Мітки вузлів можуть бути тимчасові і постійні. Тимчасова мітка може бути замінена

на іншу тимчасову, якщо буде знайдено більш короткий шлях до даного вузла. Якщо такого шляху не існує, то тимчасова мітка змінюється на постійну.

Позначимо через u_i найкоротший шлях вихідного вузла 1 до вузла i , через d_{ij} – довжину дуги (i, j) . Для вузла j визначимо мітку $[u_j, i]$ так:

$$[u_j, i] = [u_i + d_{ij}, i], d_{ij} \geq 0.$$

Ітеративна процедура алгоритму Дейкстри складається з таких кроків.

Крок 0. Вихідному вузлу (вузол 1) приписуємо мітку $[0, -]$. Вважаємо $i = 0$.

Крок i .

1. Обчислюємо тимчасові мітки $[u_i + d_{ij}, i]$ для всіх вузлів j , які можна досягти безпосередньо із вузла i і які не мають постійних міток. Якщо вузол j вже має мітку $[u_j, k]$, яка отримана від іншого вузла k , і якщо $u_i + d_{ij} < u_j$, тоді мітка $[u_j, k]$ замінюється на $[u_i + d_{ij}, i]$.

2. Якщо всі вузли мають постійні мітки, процес обчислень закінчується. У протилежному випадку обирається мітка $[u_r, s]$ з найменшим значенням відстані u_r серед усіх тимчасових міток. Вважаємо $i = r$ і повторюємо крок i .

Приклад. Знайти найкоротший ланцюг у графі (рис.3.1).

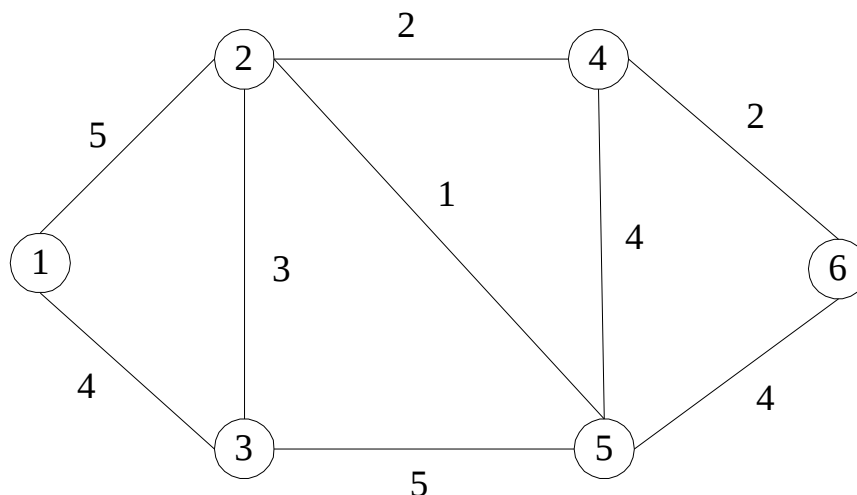


Рис.3.1. Приклад мережі для визначення найкоротшого шляху

Послідовність рішення задачі наведено в табл.. 2.1.

Таблиця 2.1

№	1	2	3	4	5	6
0	0	∞	∞	∞	∞	∞
1	0	5	4	∞	∞	∞
2	0	5	4	∞	9	∞
3	0	5	4	7	6	∞
4	0	5	4	7	6	10
5	0	5	4	7	6	9

Згідно з алгоритмом розв'язання задачі знайдено: найкоротший маршрут з вершини 1 у вершину 6 дорівнює 9.

3.3. Задача про багатополюсну мережу.

Алгоритм Флойда – Ху

Постановка задачі. Необхідно визначити найкоротші ланцюги між усіма вузлами багатополюсної мережі.

Алгоритм Флойда – Ху.

Алгоритм будується на таких положеннях. Необхідно визначити найкоротший ланцюг із вузла i у вузол k (рис. 3.2).

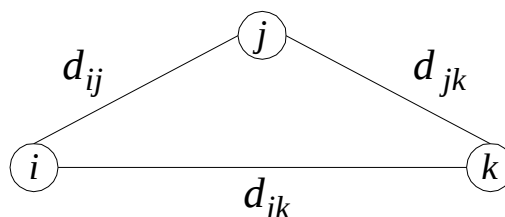


Рис. 3.2. Фрагмент мережі для визначення відстані між вузлами i і k

Спочатку ця довжина береться рівною довжині дуги (i, k) і дорівнює d_{ik} , або ∞ , якщо з'єднання немає. Потім послідовно перевіряються всі можливі проміжні вузли, розташовані між i і k . Якщо довжина ланцюга, що проходить через деякий проміжний вузол j , менша за поточне значення, то змінній d_{ik}

надається нове значення $d_{ik} = d_{ij} + d_{jk}$. Дана процедура повторюється для всіх можливих пар вузлів. Задача для n вузлів розв'язується за допомогою n ітерацій.

На кожній ітерації j відстань обчислюється так:

$$d_{ik}^j = \min [d_{ik}^{j-1}, d_{ij}^{j-1} + d_{jk}^{j-1}], \quad i \neq j \neq k.$$

На кожній ітерації будуються дві матриці. Перша з них називається **матрицею довжин** найкоротших шляхів $D^j = [d_{ik}^j]$ і означає найкоротші відстані між усіма парами вузлів. Друга матриця називається **матрицею маршрутів** $R^j = [r_{ik}^j]$ і означає проміжні вузли найкоротших ланцюгів між усіма парами вузлів. Тут r_{ik}^j – проміжний вузол найкоротшого ланцюга з i в k , обраний на j -й ітерації: $r_{ik}^j = j$, якщо $d_{ik}^{j-1} > d_{ij}^{j-1} + d_{jk}^{j-1}$; $r_{ik}^j = r_{ik}^{j-1}$, в протилежному випадку.

Крок 0. $j=0$. Будуємо початкову матрицю вартостей D^0 і початкову матрицю маршрутів R^0 . Елементи d_{ij}^0 початкової матриці вартостей дорівнюють вартості дуги (i,k) , якщо вузли i та k безпосередньо зв'язані дугою (i,k) , і ∞ в супротивному разі, всі діагональні елементи матриці D^0 дорівнюють нулю. Елементи r_{ik}^0 матриці маршрутів R^0 дорівнюють k .

Крок 1. Починаючи з $j = 1$, виділяємо в матриці D^{j-1} j -й стовпець та j -й рядок, оголошуємо їх **базовими** і виконуємо таку операцію:

$$d_{ik}^j = \min [d_{ik}^{j-1}, d_{ij}^{j-1} + d_{jk}^{j-1}].$$

Елементи r_{ik}^j матриці R^j знаходять через елементи r_{ik}^{j-1} матриці R^{j-1} за такою схемою: $r_{ik}^j = j$, якщо $d_{ik}^{j-1} > d_{ij}^{j-1} + d_{jk}^{j-1}$; $r_{ik}^j = r_{ik}^{j-1}$, в протилежному випадку.

Приклад. Для прикладу графа, який наведено при розгляді алгоритму Дейкстри, визначити найкоротші відстані між усіма парами вузлів і відповідні їм маршрути (рис. 3.3).

Крок 0

D^0

	1	2	3	4	5	6
1	—	5	4	∞	∞	∞
2	5	—	3	2	1	∞
3	4	3	—	∞	5	∞
4	∞	2	∞	—	4	2
5	∞	1	5	4	—	4
6	∞	∞	∞	2	4	—

R^0

	1	2	3	4	5	6
1	—	2	3	4	5	6
2	1	—	3	4	5	6
3	1	2	—	4	5	6
4	1	2	3	—	5	6
5	1	2	3	4	—	6
6	1	2	3	4	5	—

Крок 1

D^1

	1	2	3	4	5	6
1	—	5	4	∞	∞	∞
2	5	—	3	2	1	∞
3	4	3	—	∞	5	∞
4	∞	2	∞	—	4	2
5	∞	1	5	4	—	4
6	∞	∞	∞	2	4	—

R^1

	1	2	3	4	5	6
1	—	2	3	4	5	6
2	1	—	3	4	5	6
3	1	2	—	4	5	6
4	1	2	3	—	5	6
5	1	2	3	4	—	6
6	1	2	3	4	5	—

Крок 2

D^2

	1	2	3	4	5	6
1	—	5	4	7	6	∞
2	5	—	3	2	1	∞
3	4	3	—	5	4	∞
4	7	2	5	—	3	2
5	6	1	4	3	—	4
6	∞	∞	∞	2	4	—

R^2

	1	2	3	4	5	6
1	—	2	3	2	2	6
2	1	—	3	4	5	6
3	1	2	—	2	2	6
4	2	2	2	—	2	6
5	2	2	2	2	—	6
6	1	2	3	4	5	—

Рис. 3.3. Послідовність розв'язання задачі про багатополюсну мережу (див. також с. 46, с. 47)

Крок 3 D^3

	1	2	3	4	5	6
1	—	5	4	7	6	∞
2	5	—	3	2	1	∞
3	4	3	—	5	4	∞
4	7	2	5	—	3	2
5	6	1	4	3	—	4
6	∞	∞	∞	2	4	—

R^3

	1	2	3	4	5	6
1	—	2	3	2	2	6
2	1	—	3	4	5	6
3	1	2	—	2	2	6
4	2	2	2	—	2	6
5	2	2	2	2	—	6
6	1	2	3	4	5	—

Крок 4 D^4

	1	2	3	4	5	6
1	—	5	4	7	6	9
2	5	—	3	2	1	4
3	4	3	—	5	4	7
4	7	2	5	—	3	2
5	6	1	4	3	—	4
6	9	4	7	2	4	—

R^4

	1	2	3	4	5	6
1	—	2	3	2	2	4
2	1	—	3	4	5	4
3	1	2	—	2	2	4
4	2	2	2	—	2	6
5	2	2	2	2	—	6
6	4	4	4	4	5	—

Крок 5 D^5

	1	2	3	4	5	6
1	—	5	4	7	6	9
2	5	—	3	2	1	4
3	4	3	—	5	4	7
4	7	2	5	—	3	2
5	6	1	4	3	—	4
6	9	4	7	2	4	—

R^5

	1	2	3	4	5	6
1	—	2	3	2	2	4
2	1	—	3	4	5	4
3	1	2	—	2	2	4
4	2	2	2	—	2	6
5	2	2	2	2	—	6
6	4	4	4	4	5	—

Рис. 3.3. Продолжения

	<i>D</i>					
	1	2	3	4	5	6
1	—	5	4	7	6	9
2	5	—	3	2	1	4
3	4	3	—	5	4	7
4	7	2	5	—	3	2
5	6	1	4	3	—	4
6	9	4	7	2	4	—

	<i>R</i>					
	1	2	3	4	5	6
1	—	2	3	2	2	4
2	1	—	3	4	5	4
3	1	2	—	2	2	4
4	2	2	2	—	2	6
5	2	2	2	2	—	6
6	4	4	4	4	5	—

Рис. 3.3. Закінчення

Матриці D і R є розв'язками задачі.

3.4. Задача про оптимальне дерево-кістяк

Постановка задачі. Необхідно визначити дерево-кістяк, сумарна вартість ребер якого – оптимальна (максимальна або мінімальна).

Алгоритм побудови оптимального дерева-кістяка.

Крок 1. На множині вершин N мережі (N, S) , де S – множина ребер, вводимо дві множини: A – множина з'єднаних вузлів і $\bar{A}$ – множина вузлів які не з'єднані. Припускається, що $A \cup \bar{A} = N$ і $A \cap \bar{A} = \emptyset$. Спочатку всі вузли належать множині $\bar{A}$: $\bar{A}=N$, $A=\emptyset$.

Крок 2. Вибираємо будь-який вузол із множини A і з'єднуємо його з іншим вузлом ребром мінімальної (максимальної) вартості і два одержані вузли вилучаємо з множини $\bar{A}$ і включаємо їх у множину A . В кінці цього кроку множина A складається з двох вузлів.

Крок 3. Серед усіх ребер, які з'єднують вузли з множини $\bar{A}$ з вузлами із множини A , вибираємо ребро з оптимальною вартістю і кінцевий вузол цього ребра, який належить множині $\bar{A}$, вилучаємо з множини $\bar{A}$ і приєднуємо до множини A .

Крок 4. Якщо $\bar{A} \neq 0$, то перейти на виконання кроку 3. При $\bar{A}=0$ алгоритм закінчує роботу.

Приклад. Для прикладу графа, який наведено при розгляді алгоритму Дейкстри, побудувати мінімальне дерево-кістяк (рис.3.4).

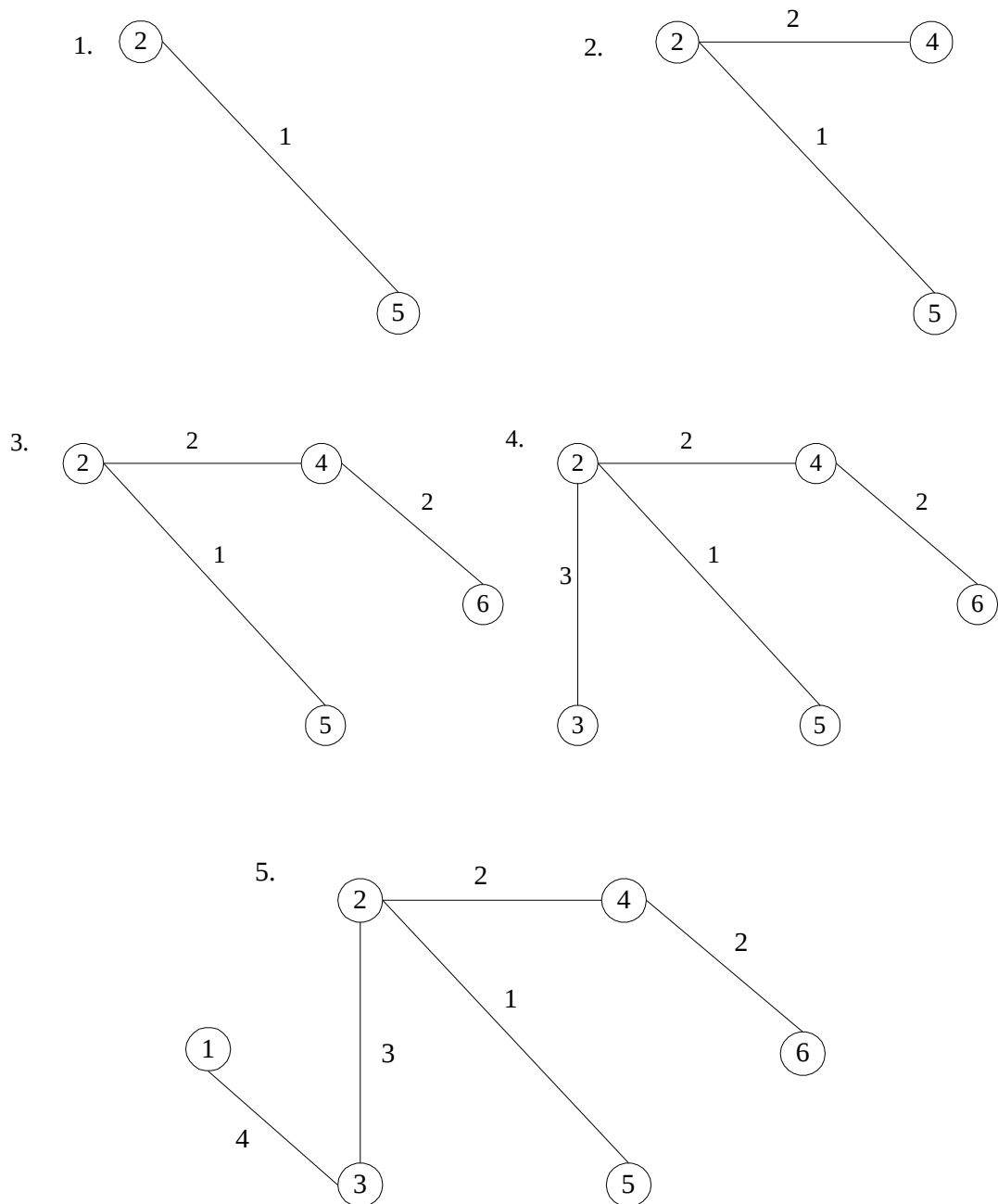


Рис.3.4. Розв'язання задачі про мінімальне дерево-кістяк

Таблиця 3.5

Споживачі Поста- чальники		B_1	B_2	B_3	B_4	U_i
		40	50	110	40	
A_1	90	² 40	⁵ 2	³ 10	¹ 40	0
A_2	80	³ 1	¹ 50	² 30	⁴ 0	-1
A_3	70	⁴ 0	² 0	¹ 70	⁵ -1	-2
V_j		2	2	3	1	

Згідно з алгоритмом розв'язання задачі знайдено: $z = 330$.

3.6. Задача про призначення. Угорський алгоритм

Постановка задачі. Необхідно n -ресурсів оптимальним чином призначити на n -об'єктів так, що кожен ресурс призначається тільки на один об'єкт і кожному об'єкту приписується тільки один ресурс.

Математична модель задачі про призначення має вигляд:

$$\sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \max (\min);$$

$$\sum_{j=1}^n x_{ij} = 1, \quad i = \overline{1, n};$$

$$\sum_{i=1}^n x_{ij} = 1, \quad j = \overline{1, n},$$

де c_{ij} – ефективність ресурсу i на об'єкті j ;

$$x_{ij} = \begin{cases} 1, & \text{якщо } i - \text{ресурс призначається на } j - \text{й об'єкт;} \\ 0 & \text{у протилежному випадку.} \end{cases}$$

Угорський алгоритм розв'язання задачі про призначення.

Метод заснований на редукції вихідної матриці і призначенні за нульовими елементами редукованої матриці. При неможливості призначення за нульовими елементами матриця перетворюється так, щоб допустимих нулів було більше, ніж в редукованій матриці.

Існує матриця $C = \| c_{ij} \|$.

Виконаємо перетворення цієї матриці:

$$a_{ij} = c_{ij} - u_{ij} - v_{ij},$$

де u_{ij} – мінімальний елемент для рядка i , а v_{ij} – мінімальний елемент для стовпця j .

$$a_{ij}x_{ij} = c_{ij}x_{ij} - u_{ij}x_{ij} - v_{ij}x_{ij},$$

$$\sum_{i=1}^n \sum_{j=1}^n a_{ij}x_{ij} = \sum_{i=1}^n \sum_{j=1}^n c_{ij}x_{ij} - \sum_{i=1}^n \sum_{j=1}^n u_{ij}x_{ij} - \sum_{i=1}^n \sum_{j=1}^n v_{ij}x_{ij}.$$

Крок 1. Перейти від початкової матриці до редукованої. Для цього від усіх елементів кожного рядка відняти мінімальний елемент цього рядка, а потім у редукованій по рядках матриці виконати таку саму операцію зі стовпцями. Після цього початкову матрицю замінити редукованою матрицею.

Крок 2. Виконати призначення. Якщо після редукції в кожному рядку i в кожному стовпці можна вибрати по одному нульовому елементу так, щоб виконувалась система обмежень, то таке призначення буде оптимальним. Якщо цього не можна зробити, то треба перейти на наступний крок.

Крок 3. Модифікація редукованої матриці. Закреслити в редукованій матриці всі стовпці і рядки, які містять хоча б один нуль, починаючи з тих рядків і стовпців, які містять максимальне число нулів, і продовжувати закреслення в такому порядку й далі. Із всіх невикреслених елементів знайти найменший, відняти його від кожного невикресленого елемента і додати його до кожного елемента, розташованого на перетині двох ліній, за допомогою яких були викреслені рядки та

стовпці. Перейти на крок 2 і якщо не вдасться виконати повне призначення, асоційоване з нулями модифікованої матриці, то знову перейти на крок 3. І так доти, доки не одержимо оптимальне призначення, асоційоване з нулями модифікованої матриці.

Якщо необхідно дослідити задачу на максимум, то всі елементи матриці слід помножити на (-1) і потім скласти з найбільшим числом матриці, таким чином, щоб матриця не вміщувала від'ємних елементів. Потім задача розв'язується, як задача мінімізації.

Приклад. Розв'язати задачу про призначення при мінімізації ЦФ (рис. 3.5).

1.					
$a_i \backslash b_j$	S_1	S_2	S_3	S_4	
W_1	2	8	12	6	
W_2	18	14	20	18	
W_3	8	10	22	14	
W_4	16	14	16	10	

2.					
$a_i \backslash b_j$	S_1	S_2	S_3	S_4	min
W_1	2	8	12	6	2
W_2	18	14	20	18	14
W_3	8	10	22	14	8
W_4	16	14	16	10	10

3.					
$a_i \backslash b_j$	S_1	S_2	S_3	S_4	
W_1	0	6	10	4	
W_2	4	0	6	4	
W_3	0	2	14	6	
W_4	6	4	6	0	
min	0	0	6	0	

4.					
$a_i \backslash b_j$	S_1	S_2	S_3	S_4	
W_1	0	6	4	4	
W_2	4	0	0	4	
W_3	0	2	8	6	
W_4	6	4	0	0	

Рис. 3.5. Послідовність розв'язання задачі про призначення (див. також с. 56)

5.

$a_i \backslash b_j$	S_1	S_2	S_3	S_4
W_1	0	6	4	4
W_2	4	0	0	4
W_3	0	2	8	6
W_4	6	4	0	0

6.

$a_i \backslash b_j$	S_1	S_2	S_3	S_4
W_1	0	4	2	2
W_2	6	0	0	4
W_3	0	0	6	4
W_4	8	4	0	0

Рис. 3.5. Закінчення

Згідно з алгоритмом розв'язування задачі знайдено оптимальні призначення: W_1-S_1 , W_2-S_3 , W_3-S_2 , W_4-S_4 .

3.7. Задача про максимальний потік

Постановка задачі. Необхідно визначити максимальний потік з джерела в стік для мережі заданої конфігурації.

Алгоритм знаходження максимального потоку.

Алгоритм ґрунтується на тому, що починаючи з джерела S , обирається будь-який ланцюг у стік T і помічаються вузли цього ланцюга. Якщо стік можна помітити, то тоді на цьому ланцюгу можна збільшити потік. Після виконання цих операцій всі мітки стираються і робляться нові мітки. Значення, які отримуються в результаті послідовних кроків, додаються. Якщо стік не можна помітити, то отриманий потік і є оптимальним.

Кожна вершина j отримає мітку або (i^+, ε_j) , або (i^-, ε_j) , де i означає номер попередньої вершини ненасиченого ланцюга від S до T , а ε_j – ціле число, або ∞ , яке означає величину можливого приросту потоку вздовж ланцюга від джерела S до даної вершини j .

Крок 1. Позначимо вершину S міткою $(0, \infty)$. Виберемо будь-яку непомічену вершину i (спочатку це джерело S). Розглянемо всі непомічені вершини j , які пов'язані дугами з вершиною i . Ті вершини j , для яких $x_{ij} < c_{ij}$, отримують мітку (i^+, ε_j) , де $\varepsilon_j = \min(\varepsilon_i, c_{ij} - x_{ij})$. Ті вершини j , для яких $x_{ji} > 0$, отримують мітку (i^-, ε_j) , де $\varepsilon_j = \min(\varepsilon_i, c_{ij} - x_{ji})$. Потім розглянемо наступну вершину i поступимо так само. Ці операції повторюються до тих пір, поки вершина T не отримувє мітку, або поки не можна буде зробити нові мітки.

Крок 2. Вершина T отримувє мітку або (j^+, ε_t) або (j^-, ε_t) . Якщо мітка буде (j^+, ε_t) , то змінимо x_{jt} на $(x_{jt} + \varepsilon_t)$, якщо ж мітка буде (j^-, ε_t) , то змінимо x_{tj} на $(x_{tj} - \varepsilon_t)$. Потім переходимо до вершини j і поступимо так само: якщо мітка буде (i^+, ε_j) , то змінимо x_{ij} на $(x_{ij} + \varepsilon_t)$, якщо ж мітка буде (i^-, ε_j) , то змінимо x_{ji} на $(x_{ji} - \varepsilon_t)$. Ці операції повторюються до тих пір поки не буде досягнена вершина S . Після цього всі мітки стираються і робиться перехід до кроку 1.

Процес розставлення міток являє собою пошук ланцюга з S в T і визначення на цьому ланцюгу значення максимального потоку, на який сумарний потік можна збільшити. Процедура завершується, коли вже неможливо продовжувати процес розставлення міток.

Приклад. Визначити максимальний потік для даної мережі (рис. 3.6).

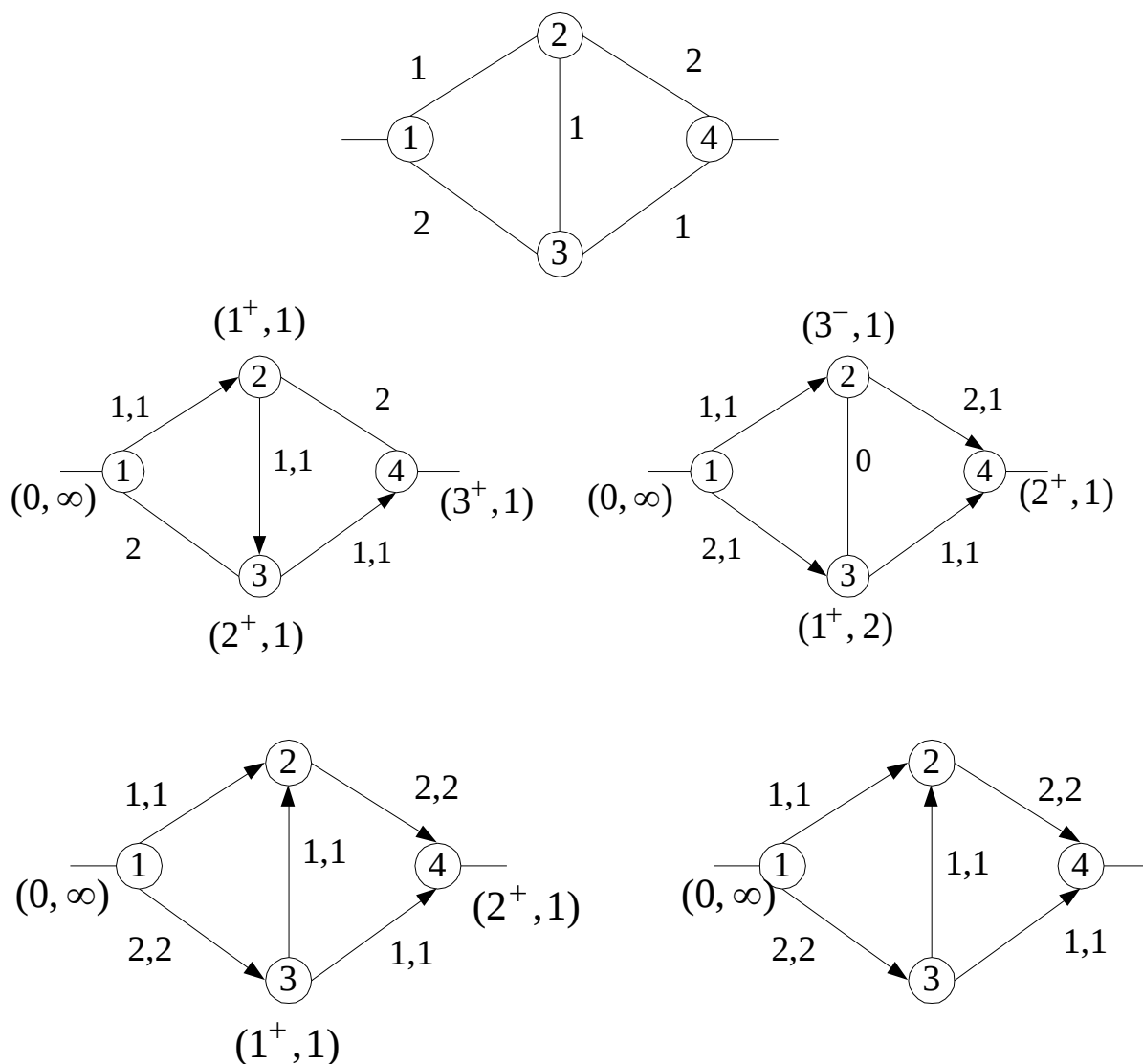


Рис. 3.6. Послідовність розв'язання задачі про максимальний потік

Згідно з алгоритмом розв'язання задачі знайдено: максимальний потік дорівнює 3.

Контрольні запитання

1. Чому симплекс-метод не є ефективним засобом для розв'язання транспортної задачі?
2. Дати визначення задачі про призначення. Чи можна її розглядати як окремий випадок транспортної задачі?

3. Дати визначення задачі про найкоротший шлях. Чи обов'язково в процесі розв'язування розглядати всі вузли?
4. В чому полягають особливості транспортної задачі з проміжними пунктами від класичної транспортної задачі?
5. Як врахувати в задачі про найкоротший шлях інтенсивність руху і ширину колії?
6. Які особливі випадки застосування транспортної задачі?
7. Чи існує зв'язок між транспортною задачею і задачею про найкоротший шлях? Якщо так, то в чому він полягає?
8. Чи може задача про мінімальне дерево-кістяк мати декілька розв'язків?
9. Пояснити принцип дії алгоритму для знаходження максимального потоку. Чому потік може змінювати свій напрям?
10. Математична модель і алгоритм розв'язання задачі про найкоротший шлях.
11. Математична модель і алгоритм розв'язання задачі про багатополіусну мережу.
12. Математична модель і алгоритм розв'язання транспортної задачі.
13. Математична модель і алгоритм розв'язання задачі про призначення.
14. Математична модель і алгоритм розв'язання задачі про оптимальне дерево-кістяк.
15. Які існують матриці для зображення графів?

Список літератури

1. Акулич И.Л. Математическое программирование в примерах и задачах: Учеб. пособие для вузов. – М.: Высш. шк., 1986. – 320 с.
2. Аттетков А.В., Галкин С.В., Зарубин В.С. Методы оптимизации: Учеб. для вузов. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2003. – 440 с.
3. Ашманов С.А., Тимохов А.В. Теория оптимизации в задачах и упражнениях. – М.: Наука, 1991. – 448 с. 4. Банди Б. Основы линейного программирования. – М.: Радио и связь, 1989. – 176 с.
4. Бондаренко М.Т. Оптимізаційні задачі в системах прийняття рішень. – Х.: УДТУР, 1998. – 216 с.
5. Волков И.К., Загоруйко Е.А. Исследование операций: Учеб. для вузов. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2002. – 436 с.
6. Гайна Г.А. Методичні вказівки до вивчення дисципліни “Методи синтезу та оптимізації”: В 4 кн. – К.: КДТУБА, 1998.
7. Зайченко Ю.П. Дослідження операцій. – К.: ЗАТ "ВІПОЛ", 2000. – 688 с.
8. Зайченко Ю.П., Шумилова С.А. Исследование операций: Сборник задач. – К.: Высш. шк., 1990. – 372 с.
9. Корниль Т.Л., Кириллова Н.А. Математическое программирование. – Х.: НТУ ХПИ, 2002. – 128 с.
10. Максимов О.В., Афанасьев М.Г. Математичне програмування. – Кривий Ріг: КРПУ, 2003. – 274 с.
11. Морозов В.В., Сухарев А.Г., Федоров В.В. Исследование операций в задачах и упражнениях : Учеб. пособие для вузов. – М.: Высш. шк., 1986. – 285 с.
12. Мур Д., Уэдерфорд Л.Д. Экономическое моделирование в Microsoft Excel. – 6-е изд. – М.: Издательский дом "Вильямс", 2004. – 1024 с. 140
13. Пантелеев А.В., Летова Т.А. Методы оптимизации в примерах и задачах. – М.: Высш. шк., 2002. – 544 с.
14. Раскин Л.Г. Математическое программирование. – Х.: НТУ ХПИ, 2002. – 126 с.
15. Реклейтис Г., Рейвиндран А., Рэгсдел К.. Оптимизация в технике: В 2 т. – М.: Мир, 1986. – Т. 1–2.
16. Таха Х. Введение в исследование операций. – М.: Издательский дом "Вильямс", 2001. – 912 с.
17. Филлипс Д., Гарсия-Диас А. Методы анализа сетей. – М.: Мир, 1984. – 496 с.
18. Черняк А.А., Новиков В.А., Мельников О.И. и др. Математика для экономистов на базе Mathcad. – СПб.: БХВ-Петербург, 2003. – 496 с.
19. Шестопад А.Ф. Елементи сітьового аналізу і лінійного програмування: Конспект лекцій. – К.: КДТУБА, 1995. – 156 с.