

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

Методичні вказівки до виконання лабораторних робіт

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, 2018 р.

Укладачі О.О. Терентьєв, доктор технічних наук, професор

Відповідальний за випуск В.М. Міхайленко, завідувач кафедри інформаційних технологій проектування та прикладної математики, доктор технічних наук, професор.

Затверджено на засіданні кафедри інформаційних технологій проектування та прикладної математики, протокол № від 2018 р.

Терентьєв О.О. Дослідження операцій: Методичні вказівки до виконання лабораторних робіт. – Київ: КНУБА, 2018. – 69 с.

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

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

ЗМІСТ

Вступ.....	2
Лабораторна робота 1 “Складання математичних моделей лінійного програмування”.....	3
Лабораторна робота 2 “Складання алгоритму розрахунку найкоротшої путі”.....	8
Лабораторна робота 3 “Складання алгоритму розрахунку потоку максимальної величини”.....	15
Лабораторна робота 4 “Розв’язання задачі про оптимальний потік”....	26
Лабораторна робота 5 “Сіткове планування робіт”.....	42
Лабораторна робота 6 “ Дослідження властивостей сітки Петрі матричним способом”.....	48
Лабораторна робота 7 “ Дослідження властивостей сітки Петрі за допомогою дерева досяжності”.....	60

ВСТУП

Досконалість процесів планування і управління в усіх галузях діяльності людини досягається за умови широкого використання економіко-математичних моделей. Використання методів моделювання з застосуванням ЕОМ дозволяють знаходити оптимальні рішення навіть при великій кількості параметрів, що визначаються, з найменшими витратами часу.

В курсі „Математичні моделі в розрахунках на ЕОМ” розглядаються алгоритми розв’язання задач оптимізації та моделювання систем. Математичне моделювання є такий метод моделювання, при якому система описується за допомогою відомих математичних функцій, рівнянь та нерівностей, але розв’язок задачі моделювання не може бути знайдений класичними аналітичними методами і знаходиться тільки в результаті реалізації відомого алгоритму, який є циклічним повторюванням певних кроків, що приводять до заданої мети.

Перша лабораторна робота курсу „Математичні моделі в розрахунках на ЕОМ” присвячена складанню та реалізації математичних моделей лінійного програмування і представляє класичний приклад математичного моделювання систем. Інша частина робіт присвячена реалізації сіткових моделей.

Сіткові моделі являються моделями, що дозволяють опис великої кількості інформації про систему найменшою кількістю змінних. Одна частина лабораторних робіт присвячена розв’язанню задач оптимізації на сіткових моделях – це задача про найкоротшу путь, задача про оптимальний потік, задача про оптимальний потік та задача про сіткове планування робіт. Інша частина лабораторних робіт присвячена розв’язанню задач з використанням сіток Петрі – це дослідження властивості зберігання, дослідження активності, задача побудови дерева досяжності. Таким чином, у курсі лабораторних робіт представлені як статичні моделі систем, так і динамічні моделі систем.

Кожний розділ містить короткі теоретичні відомості, необхідні для виконання лабораторної роботи; завдання до роботи; шість варіантів завдання; хід виконання лабораторної роботи; зміст звіту; контрольні питання до заліку лабораторної роботи; літературу, в якій можна почерпнути додаткові теоретичні відомості.

Реалізація алгоритмів, запропонованих в курсі лабораторних робіт з „Математичних моделей в розрахунках на ЕОМ”, потребує від студента досконалого знання універсальної мови програмування такої, як Pascal або C. Застосування об’єктно-орієнтованого підходу до побудови програм робить їх більш універсальними, а побудова інтерактивного інтерфейсу робить їх зручними в користуванні. Тому, окрім основної своєї мети, виконання лабораторних робіт якнайбільше сприяє удосконаленню навичок програмування студента і готує його до створення інформаційних технологій сучасного рівня.

ЛАБОРАТОРНА РОБОТА 1

“Складання математичних моделей лінійного програмування”

Мета роботи: навчитися складати та реалізовувати моделі лінійного програмування.

1. Короткі теоретичні відомості з лінійного програмування

Лінійним програмуванням називають розділ математики, в якому вивчають методи відшукування мінімуму або максимуму лінійної функції при обмеженнях на її змінні, які задані у вигляді лінійних нерівностей або рівнянь. Загальна постановка задачі лінійного програмування (ЗЛП) полягає у наступному: знайти такі значення дійсних змінних $x_1, x_2, \dots, x_n$, які задовольняють обмеженням

$$\begin{cases} a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n \leq b_i, i = 1 \dots k \\ a_{j1}x_1 + a_{j2}x_2 + \dots + a_{jn}x_n \geq b_j, i = k + 1 \dots m \\ a_{q1}x_1 + a_{q2}x_2 + \dots + a_{qn}x_n \leq b_q, l = m + 1 \dots h \\ x_i \geq 0, i = 1, \dots, n \end{cases} \quad (1.1)$$

і забезпечують мінімальне (максимальне) значення цільової функції

$$f(x) = c_1x_1 + c_2x_2 + \dots + c_nx_n \rightarrow \min(\max) \quad (1.2)$$

Для того, щоб побудувати математичну модель лінійного програмування потрібно:

- 1) визначити змінні моделі, які являються об'єктом пошуку;
- 2) сформулювати ціль пошуку;
- 3) сформулювати обмеження, яким повинні задовольняти змінні моделі.

Розглянемо процес побудови моделі на прикладі виробничого планування.

Приклад 1. Для деякої одиниці виробничого процесу (заводу, цеху) потрібно скласти план виробництва n видів продукції. У виробництві використовується m видів ресурсів. Використання кожного ресурсу обмежено його наявною кількістю b_i ($i=1, \dots, m$). Для випуску одиниці продукції j -ого виду витрачається a_{ij} ($i=1, \dots, m, j=1, \dots, n$) одиниць ресурсу. Відома ціна c_j ($j=1, \dots, n$) одиниці продукції. Потрібно знайти план виробництва, що забезпечує максимальний прибуток від виробництва усіх видів продукції.

Об'єктом пошуку в даній задачі являється план виробництва $X=(x_1, x_2, \dots, x_n)$, де x_i – невідома кількість продукції i -ого виду, яка виробляється. Ціллю пошуку являється досягнення максимального сумарного прибутку. Оскільки прибуток від випуску i -ої продукції знаходиться як добуток відомої ціни одиниці продукції і кількості продукції, то цільова функція має вид:

$$f(x) = c_1x_1 + c_2x_2 + \dots + c_nx_n \rightarrow \max .$$

Використання ресурсів обмежене їх кількістю, що математично формулюється у вигляді наступних нерівностей:

$$a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n \leq b_i, i = 1 \dots m$$

Оскільки кількість продукції, яка виробляється, не може бути від'ємною, додаються очевидні обмеження на змінні моделі:

$$x_i \geq 0, i = 1, \dots, n.$$

Після таких міркувань отримуємо математичну модель виробничого планування, що має вигляд задачі лінійного програмування (1.1), (1.2):

$$\begin{cases} a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n \leq b_i, i = 1 \dots m, \\ x_i \geq 0, i = 1, \dots, n \end{cases}$$

$$f(x) = c_1x_1 + c_2x_2 + \dots + c_nx_n \rightarrow \max.$$

Розв'язання ЗЛП знаходиться шляхом перетворювання симплекс-таблиць [1]. Реалізація цього способу універсальними мовами програмування призводить до накопичування похибок в ході перетворювань симплекс-таблиць. Тому недосвідченому програмісту для реалізації ЗЛП рекомендується використовувати спеціальне програмне забезпечення Mathcad. Звичайна ЗЛП розв'язується за допомогою Mathcad-функцій *maximize(f,x)* або *minimize(f,x)*, аргументами яких являються ім'я цільової функції *f* та вектор змінних *x* [3]. Детальне пояснення використання цих функцій розглянемо на прикладі.

Приклад 2. Нехай потрібно розв'язати наступну ЗЛП:

$$\begin{cases} x_1 + x_2 \leq 3000 \\ x_3 + x_4 \leq 2500 \\ x_5 + x_6 \leq 2800 \\ x_1 + \frac{1}{2}x_3 + \frac{1}{3}x_5 = 100 \\ \frac{1}{3}x_2 + \frac{1}{4}x_4 + x_6 = 500 \\ x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0, x_5 \geq 0, x_6 \geq 0 \end{cases}$$

$$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \rightarrow \min$$

Присвоїмо вбудованій Mathcad-змінній ORIGIN, яка відповідає за нумерацію компонентів вектора, значення 1:

$$\text{ORIGIN}:=1$$

Далі введемо початкові довільні значення вектора змінних *x*:

$$x := \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

Введемо цільову функцію:

$$f(x) := x_1 + x_2 + x_3 + x_4 + x_5 + x_6.$$

Сформуємо блок чисельного розв'язання Given: на початку його напишемо слово Given, нижче напишемо всі нерівності-обмеження, ще нижче напишемо ту функцію, розрахунок якої хочемо отримати - *Maximize(f,x)* або *Minimize(f,x)*. Отримаємо наступний результат:

Given

$$x_1 + x_2 \leq 3000$$

$$x_3 + x_4 \leq 2500$$

$$x_5 + x_6 \leq 2800$$

$$x_1 + \frac{1}{2}x_3 + \frac{1}{3}x_5 = 100$$

$$\frac{1}{3}x_2 + \frac{1}{4}x_4 + x_6 = 500$$

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

$$\text{Minimize}(f, x) = \begin{pmatrix} 100 \\ 0 \\ 0 \\ 0 \\ 0 \\ 500 \end{pmatrix}$$

Дійсно, значення $x_1=100$, $x_6=500$, інші $x_i=0$ забезпечують виконання, по-перше, всіх обмежень і, по-друге, функція $f(x)$ приймає найменше значення $100+500=600$. Якби, наприклад, $x_1=80$, $x_3=40$, $x_6=500$, то обмеження виконані, проте значення цільової функції стає більше мінімального: $80+40+500=620$. Аналогічно, якщо трохи зменшити значення шостої змінної, наприклад $x_6=495$, то для виконання обмежень необхідно збільшити значення іншої змінної, наприклад, $x_4=20$. Значення цільової функції знову збільшується: $100+20+495=615$.

Таким чином робимо висновок про правильність розрахунку Mathcad-програмою.

Зауваження: ЗЛП може не мати розв'язку за таких причин:

- 1) область обмежень пуста;
- 2) значення цільової функції поліпшується до нескінченності, оскільки область обмежень простирається у напрямку поліпшення цільової функції до нескінченності.

Mathcad-програма розв'язання ЗЛП складається з наступних дій:

- 1) ввести нумерацію компонентів вектора з першого, скориставшись змінною ORIGIN;
- 2) присвоїти початкові значення вектора змінних x ;
- 3) ввести цільову функцію $f(x)$;
- 4) написати ключове слово *Given*;
- 5) задати нерівності-обмеження на вектор змінних;
- 6) написати, що потрібно розрахувати - функцію *Maximize(f,x)* або *Minimize(f,x)* зі звичайним знаком „=” після неї.

Аналіз чутливості ЗЛП – це дослідження впливу змінювання коефіцієнтів a_{ij} , коефіцієнтів цільової функції c_i і обмежень b_i на оптимальний розв'язок. Якщо при невеличкому (5%÷10%) змінюванні коефіцієнтів a_{ij} (коефіцієнтів цільової функції c_i і обмежень b_i) спостерігається таке ж невеличке (5%÷10%)

змінювання значення оптимального розв'язку, то ЗЛП являється нечутливою. Якщо ж при невеличкому ($5\% \div 10\%$) змінюванні коефіцієнтів a_{ij} (коефіцієнтів цільової функції c_i і обмежень b_i) спостерігається значне ($50\% \div 100\%$) змінювання значення оптимального розв'язку, то ЗЛП являється чутливою. Дослідження чутливості являється важливим у зв'язку з неточністю початкових даних.

2. Завдання до роботи

Для даного варіанту сітки:

- 1) скласти модель за текстом варіанту завдання (см. п.3) у вигляді задачі лінійного програмування;
- 2) скласти та реалізувати на ЕОМ алгоритм розв'язання ЗЛП;
- 3) перевірити складений алгоритм на простій задачі, для якої відомий розв'язок;
- 4) зробити розрахунки за даними варіанту завдання.

При складанні алгоритму передбачити можливість введення параметрів моделі користувачем.

3. Варіанти

- 1) Задача вибору засобів доставки.

Для перевезення p видів вантажів використовують кораблі n типів, причому a_k - кількість вантажів кожного виду, $k=1, \dots, p$, q_j - кількість кораблів типу j , c_j - витрати при використанні одного корабля типа j . Кожний корабель має m ємностей (трюми, палуби), d_{ij} - вантажопідйомність ємності i , $i=1, \dots, m$, на кораблі типа j . Потрібно вибрати найбільш економічний комплекс засобів доставки і план вантаження кораблів.

- 2) Задача про використання виробничого обладнання.

Потрібно виготовити чотири види виробів A_i у кількості від a_i до b_i ($i=1,2,3,4$). Собівартість одного виробу складає c_i . Кожний виріб A_i потребує t_{ij} хвилин робочого часу на кожному із станків M_j ($j=1,2,3$), загальний час використання яких повинен знаходитись у межах від n_j до m_j хвилин. Потрібно знайти план виробництва з мінімальною собівартістю.

- 3) Задача виробничого планування.

У виробництві використовують m складових (сировина, виробничі фонди та ін.). Існує n технологічних способів організації цього виробництва, a_{ij} - витрати ($a_{ij} \geq 0$) або прибуток ($a_{ij} \leq 0$) i -ої складової при роботі по j -ому технологічному способу з одиничною інтенсивністю, b_i - загальна кількість i -ої складової, c_j - доход ($c_j \geq 0$) або збитки ($c_j \leq 0$) від використання j -ої технології з одиничною інтенсивністю. Потрібно знайти план виробництва, що забезпечує максимальний сумарний доход від усіх способів виробництва.

- 4) Планування розкрою.

Для заготовок у вигляді стержня довжини l кожний є варіанти розкрою Z_i , $i=1,2,3,4,5,6$. Потрібно виробити a_j частин T_j довжини l_j , $j=1,2,3,4$. Відомо, що варіант розкрою Z_i забезпечує k_{ij} частин T_j . (Умова $k_{i1} \cdot l_1 + k_{i2} \cdot l_2 + k_{i3} \cdot l_3 + k_{i4} \cdot l_4 \leq l$ міститься у визначенні „варіант розкрою” і не повинна використовуватись як

умова ЗЛП). Потрібно виготовити замовлену кількість частин із мінімальної кількості заготовок.

5) Задача про покриття.

Для будівництва ГЕС розглядаються 6 місць розташування Z_i . Кожна ГЕС Z_i при розташуванні в i -ому місці могла б обслужити $k_{ij}=1$ або не обслужити $k_{ij}=0$ деякі з чотирьох регіонів. Після спорудження мінімальної кількості ГЕС кожний із чотирьох регіонів повинен покриватися хоча б один раз.

6) Задача про суміші.

Потрібно виготовити сплав із трьох металів M_i . Кожний метал має густину a_{i1} , вміст вуглецю a_{i2} та вміст фосфору a_{i3} , і може бути використані у кількості від b_i до c_i і коштує g_i гривень за кілограм. Сплав повинен мати густину в межах від n_1 до m_1 , вміст вуглецю від n_2 до m_2 та вміст фосфору від n_3 до m_3 . Сплав потрібно виробити якнайдешевше у кількості k кілограмів.

4. Хід виконання роботи

- 1) уважно розібрати теоретичні відомості з лінійного програмування;
- 2) разом з викладачем вибрати варіант завдання;
- 3) за текстом завдання скласти математичну модель задачі;
- 4) скласти алгоритм розв'язання задачі лінійного програмування за допомогою програмного забезпечення Mathcad;
- 5) знайти розв'язок задачі лінійного програмування та дослідити його чутливість;
- 6) проаналізувати результати і зробити висновки щодо правильності отриманих розрахунків та реалізованого алгоритму;
- 7) оформити звіт.

5. Зміст звіту

- 1) завдання до роботи;
- 2) математична модель побудованої задачі лінійного програмування;
- 3) лістинг програми;
- 4) результати розрахунків;
- 5) результати аналізу чутливості моделі;
- 6) висновки.

6. Контрольні питання

- 1) У чому полягає задача лінійного програмування?
- 2) Дайте математичне формулювання задачі лінійного програмування?
- 3) Що таке цільова функція?
- 4) Як скласти математичну модель лінійного програмування?
- 5) За допомогою яких Mathcad-функції розв'язується задача лінійного програмування?
- 6) Як знаходиться розв'язок звичайної задачі лінійного програмування у програмному забезпеченні Mathcad?
- 7) Коли задача лінійного програмування не має розв'язку?
- 8) У чому полягає аналіз чутливості моделі лінійного програмування?

7. Література

1. Деордица Ю.С., Нефедов Ю.М., Исследование операций в планировании и управлении: Уч. пособие, К.: Выща школа, 1991. – 270с.
2. Херхагер М. Партолль Х. Mathcad 2000. – К.: “Ирина”, ВНУ, 2000. – 414 с.

ЛАБОРАТОРНА РОБОТА 2

“Складання алгоритму розрахунку найкоротшої путі”

Мета роботи: навчитися складати та реалізовувати алгоритм розрахунку найкоротшої путі.

1. Короткі теоретичні відомості з алгоритму розв’язання задачі про найкоротшу путь

Граф є заданим, якщо задана множина вершин I та множина дуг U , причому кожна дуга є упорядкованою парою (i, j) елементів, що зв’язують ці вершини.

Сіткою називається граф, елементам якого поставлені у відповідність деякі числові значення.

Сіткові моделі – це моделі об’єктів, побудовані з використанням сіток.

Нехай дана множина вершин I та множина дуг U , що з’єднують ці вершини. Для кожної дуги (i, j) задане число c_{ij} , яке означає вартість використання даної дуги. Число c_{ij} може характеризувати відстань від i до j , вартість слідування по маршруту від i до j , швидкість руху на маршруті від i до j і т. п. *Путтю* називається послідовність дуг P , початок кожної з яких співпадає з кінцем попередньої. *Задача про найкоротшу путь* полягає у тому, щоб для даної вершини-початку та для даної вершини-кінця відшукати путь, довжина якої є найкоротшою: $\sum_{(i,j) \in P} c_{ij} \rightarrow \min$.

Існує багато способів розв’язання цієї задачі. Розглянемо, наприклад, *алгоритм Мінті*, який використовується для двонаправлених дуг. Тобто, якщо дозволяється рух у напрямку від i до j , то дозволяється також рух у напрямку від j до i .

Алгоритм Мінті розв’язання задачі про найкоротшу путь полягає у наступних кроках:

- Крок 0. Відмітити вершину-початок міткою $m_{start}=0$, та присвоїти поточним значенням c_{ij} початкові значення : $c_{ij}'=c_{ij}$. Перейти до кроку 1.
- Крок 1. Відмітити міткою $m_j=i$ ті (невідмічені) вершини j , для яких $c_{ij}'=0$. Якщо вершина j є вершина-кінець, то кінець алгоритму. Інакше виконувати крок 2.
- Крок 2. Для всіх відмічених вершин i знайти всі дуги, що з’єднують відмічену вершину i та невідмічену ще вершину j , і обчислити

значення величини $\delta = \min c_{ij}$. Зменшити всі поточні значення c_{ij} на величину δ : $c_{ij} = c_{ij} - \delta$. Перейти до кроку 1.

Найкоротша путь визначається за мітками m_j , починаючи з кінцевої вершини: з вершини *finish* найкоротша путь проходить через вершину з номером $j = m_{\text{finish}}$, потім через вершину з номером m_j і т.д. до вершини *start*.

Проілюструємо алгоритм Мінті на прикладі.

Приклад 1. Сім вершин 1, 2, 3, 4, 5, 6, 7 з'єднані дванадцятьма дугами 12, 13, 23, 24, 25, 34, 35, 45, 46, 56, 57 та 67 (рисунок 2.1). Відомі значення відстаней:

$c_{12}=9$, $c_{23}=16$, $c_{25}=5$, $c_{35}=2$, $c_{46}=13$, $c_{57}=7$,
 $c_{13}=12$, $c_{24}=32$, $c_{34}=6$, $c_{45}=15$, $c_{56}=1$, $c_{67}=21$.

Потрібно знайти найкоротшу путь від вершини 1 до вершини 7.

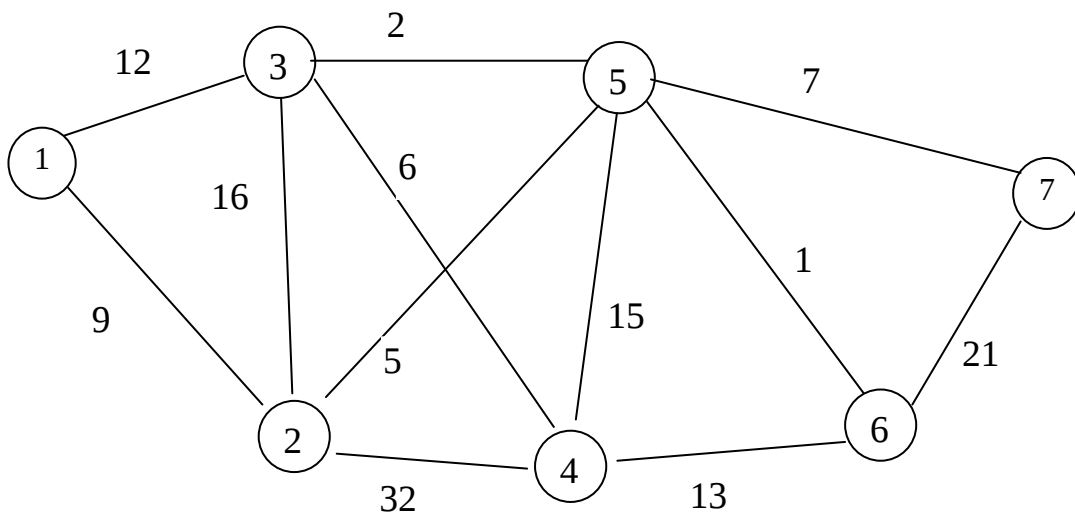


Рисунок 2.1. Сітка до прикладу 1

За алгоритмом Мінті розрахунок найкоротшої путі складається з таких дій (рисунок 2.2):

0) Відмічаємо вершину-початок $m_1 := 0$ (всі інші вершини є ще невідміченими). Присвоюємо поточним значенням c_{ij} початкові значення: $c_{ij} = c_{ij}$. Символ ` надалі опускаємо для простоти викладення.

1) Знаходимо дуги, що з'єднують вершину 1 та невідмічені ще вершини – це дуги 12 та 13. Обчислюємо значення величини $\delta = \min\{c_{12}, c_{13}\} = 9$. Розраховуємо поточні значення величин $c_{12} := 9 - 9 = 0$; $c_{13} := 12 - 9 = 3$. Відмічаємо вершину 2 міткою $m_2 := 1$ (оскільки $c_{12} = 0$).

2) Знаходимо дуги, що з'єднують відмічені вершини 1 та 2 з невідміченими ще вершинами – це дуги 13, 24, 25 та 23. Обчислюємо значення величини $\delta = \min\{c_{13}, c_{24}, c_{25}, c_{23}\} = 3$ та розраховуємо поточні значення величин $c_{13} := 3 - 3 = 0$; $c_{24} := 32 - 3 = 29$; $c_{25} := 5 - 3 = 2$; $c_{23} := 16 - 3 = 13$. Відмічаємо вершину 3 міткою $m_3 := 1$.

3) Продовжуємо аналогічно тепер уже для відмічених вершин 1, 2 та 3: $\delta = \min\{c_{35}, c_{24}, c_{25}, c_{34}\} = 2$; $c_{35} := 2 - 2 = 0$; $c_{24} := 29 - 2 = 27$; $c_{25} := 2 - 2 = 0$; $c_{34} := 6 - 2 = 4$. Відмічаємо вершину 5 міткою $m_5 := 3$ (або 2, оскільки також $c_{25} = 0$).

4) До відмічених вершин приєдналась вершина 5, тому $\delta = \min\{c_{57}, c_{56}, c_{24}, c_{34}\} = 1$; $c_{57} := 7 - 1 = 6$; $c_{56} := 1 - 1 = 0$; $c_{24} := 27 - 1 = 26$; $c_{34} := 4 - 1 = 3$. Відмічаємо вершину 6 міткою $m_6 := 5$.

5) Розраховуємо $\delta = \min\{c_{57}, c_{67}, c_{24}, c_{34}, c_{64}\} = 3$; $c_{57} := 6 - 3 = 3$; $c_{67} := 21 - 3 = 18$; $c_{24} := 26 - 3 = 23$; $c_{34} := 3 - 3 = 0$; $c_{64} := 13 - 3 = 10$. Відмічаємо вершину 4 міткою $m_4 := 3$.

6) Розраховуємо $\delta = \min\{c_{57}, c_{67}\} = 3$; $c_{57} := 3 - 3 = 0$; $c_{67} := 18 - 3 = 15$. Відмічаємо вершину 7, яка є вершиною-кінцем, міткою $m_7 := 5$ і припиняємо розрахунки.

Найкоротша путь проходить через вершини 1-3-5-7 або через вершини 1-2-5-7 (на рисунку 2.2 зображений сірим кольором тільки перший із них). Довжина найкоротшої путі складає $12 + 2 + 7 = 21$ (або $9 + 5 + 7 = 21$).

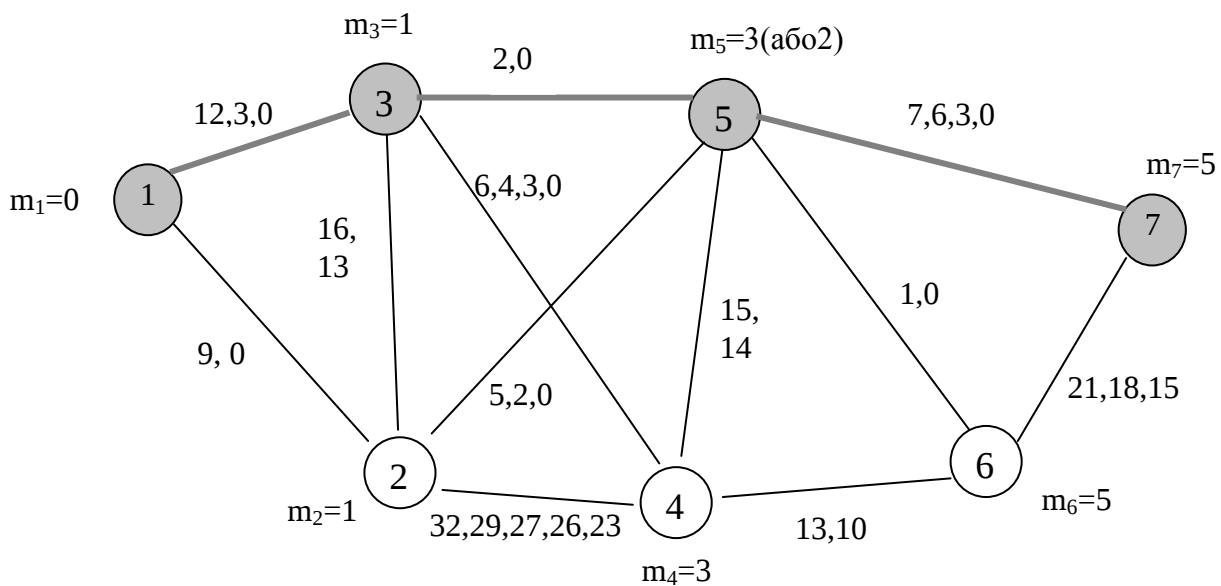


Рисунок 2.2. Розв'язання задачі про найкоротший путь алгоритмом Мінті

При реалізації алгоритму Мінті на ЕОМ найскладнішим виявляється визначення дуг, які з'єднують відмічені та невідмічені вершини.

Алгоритм Дейкстри використовується для сіток з однонаправленими дугами і дозволяє знаходити не тільки найкоротшу путь від вершини-початку до вершини-кінця, але й всі найкоротші путі від вершини 1 до інших вершин сітки.

Нехай дана сітка (I, U, c) . Вершину-початок путі позначимо як вершину 1. Для кожної вершини $i \in I$ введемо величини d_i , рівні довжині найкоротшої путі з вершини 1 до вершини i уздовж відмічених вершин.

Алгоритм Дейкстри розв'язання задачі про найкоротшу путь полягає у наступних кроках:

Крок 0. Відмітити вершину 1 і присвоїти величині $d_1 = 0$, а всім іншим величинам $d_i = 10000$ (або інше велике число, більше за всі задані значення c_{ij}). Запам'ятати номер останньої відміченої вершини: $k := 1$. Перейти до кроку 1.

Крок 1. Перерахувати значення величин d_i за формулою $d_i = \min\{d_i, d_k + c_{ki}\}$. Відмітити ту вершину j з невідмічених ще

вершин, що значення величини d_j є найменшим серед усіх невідмічених вершин, і відмітити дугу (k,j) , яка веде до цієї вершини. Запам'ятати номер $k:=j$ останньої відміченої вершини. Якщо всі $d_i=10000$, то до вершин, що залишились, путі з вершини 1 не має.

Крок 2. Якщо всі вершини є відміченими, то кінець алгоритму. Інакше перейти до кроку 1.

Відмічені дуги створюють дерево дуг з коренем у вершині 1 – дерево найкоротших путей із вершини 1 до всіх інших вершин сітки.

Проілюструємо алгоритм Дейкстри на прикладі.

Приклад 2. Вісім вершин 1, 2, 3, 4, 5, 6, 7, 8 з'єднані п'ятнадцятьма дугами 12, 13, 23, 24, 25, 34, 36, 45, 46, 47, 57, 58, 67, 68 та 78 (рисунок 2.3). Відомі значення відстаней:

$c_{12}=8,$	$c_{24}=6,$	$c_{36}=1,$	$c_{47}=3,$	$c_{67}=4,$
$c_{13}=15,$	$c_{25}=14,$	$c_{45}=17,$	$c_{57}=4,$	$c_{68}=20,$
$c_{23}=3,$	$c_{34}=11,$	$c_{46}=2,$	$c_{58}=22,$	$c_{78}=5,$

Потрібно знайти найкоротшу путь від вершини 1 до вершини 8, а також до усіх інших вершин сітки.

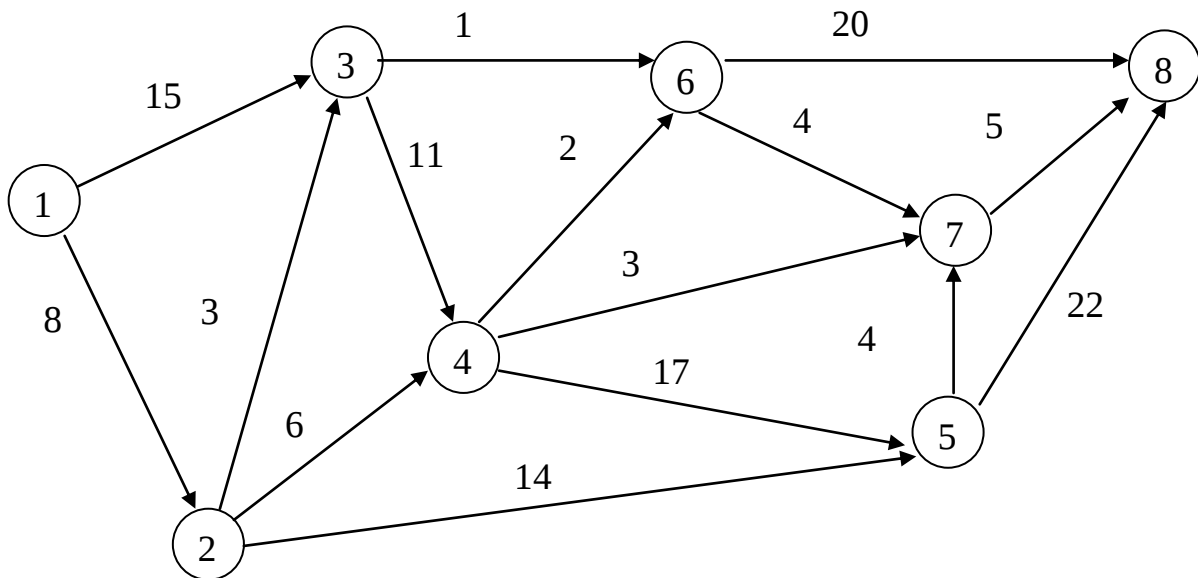


Рисунок 2.3. Сітка до прикладу 2

За алгоритмом Дейкстри розрахунок найкоротшої путі складається з таких дій (рисунок 2.4):

0) Відмічаємо вершину 1 (сірим кольором на рисунку 2.4). Присвоюємо $d_1=0$, а всім іншим величинам $d_i=1000$. Запам'ятовуємо номер останньої відміченої вершини $k=1$.

1) Знаходимо дуги, що з'єднують вершину 1 з іншими вершинами сітки – це дуги 12 та 13. Перераховуємо значення величин d_2 та d_3 : $d_2 = \min\{1000, 0+8\} = 8$, $d_3 = \min\{1000, 0+15\} = 15$. З усіх невідмічених ще вершин знаходимо вершину, для якої значення d_i найменше і відмічаємо її – це вершина

2. Відмічаємо також дугу, що забезпечила це значення d_i – це дуга 12. Запам'ятовуємо номер останньої відміченої вершини – номер 2.

2) Знаходимо дуги, що з'єднують останню відмічену вершину – це вершина 2 - з іншими вершинами сітки – це дуги 23, 24 та 25. Перераховуємо значення величин d_3 , d_4 та d_5 : $d_3 = \min\{15, 8 + 3\} = 11$, $d_4 = \min\{1000, 8 + 6\} = 14$, $d_5 = \min\{1000, 8 + 14\} = 22$. З усіх невідмічених ще вершин знаходимо вершину, для якої значення d_i найменше і відмічаємо її – це вершина 3. Відмічаємо також дугу, що забезпечила це значення d_i – це дуга 23. Запам'ятовуємо номер останньої відміченої вершини – номер 3.

3) Знаходимо дуги, що з'єднують вершину 3 з іншими вершинами сітки – це дуги 34, та 36. Перераховуємо значення величин d_4 та d_6 : $d_4 = \min\{14, 11 + 11\} = 14$, $d_6 = \min\{1000, 11 + 1\} = 12$. З усіх невідмічених вершин знаходимо вершину, для якої значення d_i найменше і відмічаємо її – це вершина 6. Відмічаємо також дугу, що забезпечила це значення d_i – це дуга 36. Запам'ятовуємо номер останньої відміченої вершини – номер 6.

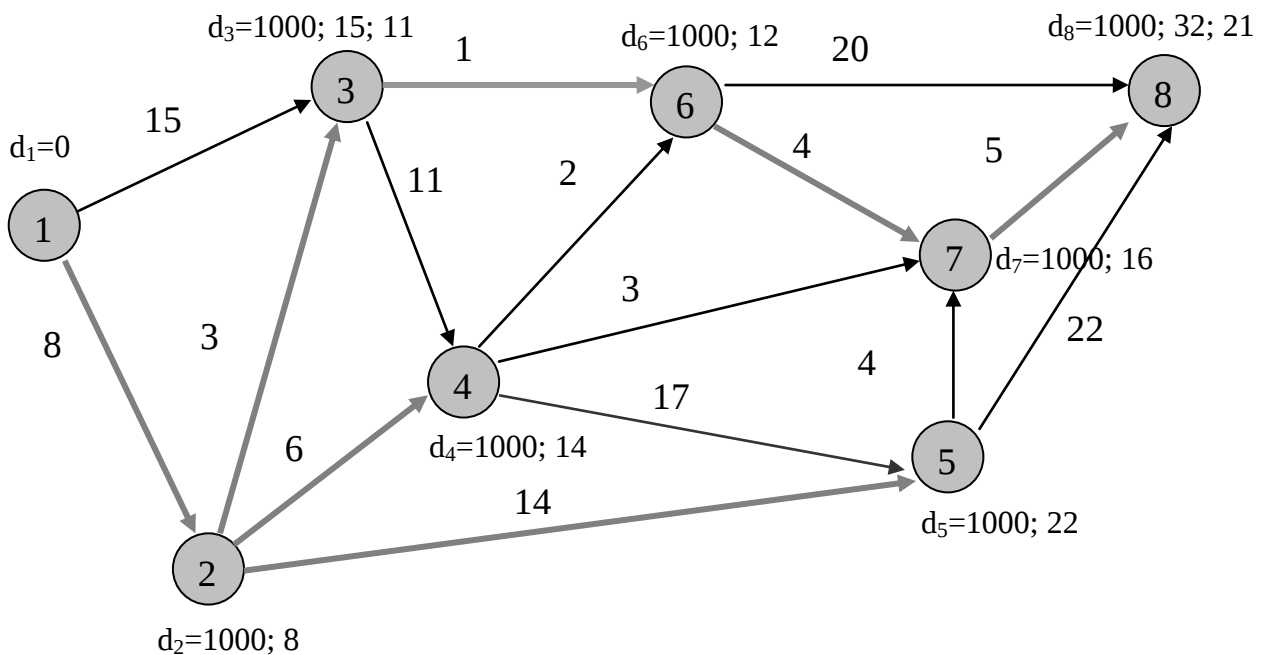


Рисунок 2.4. Розв'язання задачі про найкоротшу путь алгоритмом Дейкстри

4) Знаходимо дуги, що з'єднують вершину 6 з іншими вершинами сітки – це дуги 67, та 68. Перераховуємо значення величин d_7 та d_8 : $d_7 = \min\{1000, 12 + 4\} = 16$, $d_8 = \min\{1000, 12 + 20\} = 32$. Відмічаємо вершину 4, оскільки вона має найменше значення величини d_i . Відмічаємо також дугу, що забезпечила це значення – це дуга 24. Запам'ятовуємо номер останньої відміченої вершини – номер 4.

5) Знаходимо дуги, що з'єднують вершину 4 з іншими вершинами сітки – це дуги 45 та 47. Перераховуємо значення величин d_7 та d_8 : $d_5 = \min\{22, 14 + 17\} = 22$, $d_7 = \min\{16, 14 + 3\} = 16$. Відмічаємо вершину 7, оскільки вона має найменше значення величини d_i . Відмічаємо також дугу 67, що

забезпечила це значення. Запам'ятовуємо номер останньої відміченої вершини – номер 7.

б) Знаходимо дуги, що з'єднують вершину 7 з іншими вершинами сітки – це дуга 78. Перераховуємо значення величини d_8 : $d_8 = \min\{32, 16+5\} = 21$. Відмічаємо вершину 8, відмічаємо також дугу 78. Запам'ятовуємо номер останньої відміченої вершини – номер 8.

Отже, найкоротша путь проходить через вершини 1-2-3-6-7-8 (на рисунку 1.4 зображений сірим кольором). Довжина найкоротшої путі складає $d_8=21$ ($8+6+3+5=21$). Дерево відмічених дуг з коренем у вершині 1 є множиною найкоротших путей з вершини 1 до усіх інших вершин сітки, а значення величин d_i указують на довжини цих путей. Так, наприклад, найкоротша путь з вершини 1 до вершини 5 складається з дуг 12 та 25, а довжина цього путі рівна $d_5=22$.

При реалізації алгоритму Дейкстри на ЕОМ зручним виявляється, що перерахунок поточних значень d_i виконується тільки для дуг, які виходять з останньої відміченої вершини. Деякі труднощі виникають при визначенні дуги, яку потрібно відмітити.

2. Завдання до роботи

Для даного варіанту сітки скласти алгоритм відшукування найкоротшої путі (Мінті або Дейкстри). Розрахувати за допомогою складеного алгоритму найкоротші путі для декількох пар вершин-початків та вершин-кінців. При складанні алгоритму передбачити можливість зміни довжини будь-якої дуги.

3. Варіанти

1) Тринадцять вершин з'єднані двонаправленими дугами, довжини яких складають: $c_{1,3}=4$, $c_{1,5}=12$, $c_{1,6}=30$, $c_{2,4}=22$, $c_{2,5}=13$, $c_{2,7}=15$, $c_{2,12}=32$, $c_{3,4}=45$, $c_{3,5}=20$, $c_{3,8}=1$, $c_{3,10}=25$, $c_{4,6}=62$, $c_{4,8}=2$, $c_{5,6}=13$, $c_{5,11}=10$, $c_{6,7}=1$, $c_{6,9}=21$, $c_{7,8}=17$, $c_{7,11}=8$, $c_{7,13}=9$, $c_{8,12}=11$, $c_{8,13}=28$, $c_{9,10}=31$, $c_{9,12}=12$, $c_{9,13}=5$, $c_{10,12}=14$, $c_{11,12}=23$, $c_{11,13}=17$, $c_{12,13}=6$.

2) Дванадцять вершин з'єднані двонаправленими дугами, довжини яких складають: $c_{1,2}=3$, $c_{1,3}=8$, $c_{1,5}=6$, $c_{1,7}=12$, $c_{1,10}=45$, $c_{2,3}=17$, $c_{2,9}=32$, $c_{3,5}=35$, $c_{3,12}=70$, $c_{4,5}=1$, $c_{4,8}=25$, $c_{4,11}=62$, $c_{4,12}=56$, $c_{5,6}=3$, $c_{5,11}=90$, $c_{6,7}=1$, $c_{6,9}=21$, $c_{7,8}=7$, $c_{7,11}=64$, $c_{8,12}=9$, $c_{9,10}=1$, $c_{9,11}=8$, $c_{9,12}=3$, $c_{10,11}=14$, $c_{11,12}=23$.

3) Чотирнадцять вершин з'єднані двонаправленими дугами, довжини яких складають: $c_{1,2}=1$, $c_{1,3}=12$, $c_{1,4}=7$, $c_{1,7}=22$, $c_{1,9}=73$, $c_{1,12}=15$, $c_{2,4}=19$, $c_{2,6}=25$, $c_{2,8}=40$, $c_{2,9}=100$, $c_{3,8}=35$, $c_{3,10}=62$, $c_{4,6}=2$, $c_{4,8}=63$, $c_{5,10}=56$, $c_{6,7}=11$, $c_{6,9}=41$, $c_{7,8}=17$, $c_{7,10}=22$, $c_{7,14}=34$, $c_{8,12}=31$, $c_{8,14}=88$, $c_{9,11}=61$, $c_{9,12}=32$, $c_{10,12}=66$, $c_{10,14}=24$, $c_{11,14}=73$, $c_{12,13}=17$, $c_{12,14}=16$, $c_{13,14}=76$.

4) Одинадцять вершин з'єднані однонаправленими дугами, довжини яких складають: $c_{1,2}=25$, $c_{1,3}=3$, $c_{1,4}=12$, $c_{1,6}=8$, $c_{1,8}=31$, $c_{1,9}=9$, $c_{2,4}=18$, $c_{2,5}=6$, $c_{2,7}=29$, $c_{2,10}=15$, $c_{3,4}=19$, $c_{3,7}=50$, $c_{3,8}=45$, $c_{3,10}=34$, $c_{4,5}=31$, $c_{4,6}=33$, $c_{4,7}=26$, $c_{4,9}=18$, $c_{4,10}=51$, $c_{5,6}=17$, $c_{5,8}=24$, $c_{5,11}=64$, $c_{6,8}=52$, $c_{6,9}=49$, $c_{7,11}=21$, $c_{8,9}=38$, $c_{8,10}=59$, $c_{9,11}=34$, $c_{9,10}=27$, $c_{10,11}=9$.

5) Дванадцять вершин з'єднані однонаправленими дугами, довжини яких складають: $c_{1,2}=3$, $c_{1,3}=28$, $c_{1,5}=16$, $c_{1,7}=19$, $c_{1,10}=45$, $c_{2,3}=17$, $c_{2,9}=36$, $c_{3,5}=25$, $c_{3,12}=63$, $c_{4,5}=15$, $c_{4,8}=45$, $c_{4,11}=42$, $c_{4,10}=56$, $c_{5,6}=3$, $c_{5,12}=30$, $c_{6,7}=1$, $c_{6,9}=21$, $c_{7,8}=7$, $c_{7,11}=44$, $c_{8,12}=39$, $c_{9,10}=19$, $c_{9,11}=28$, $c_{9,12}=13$, $c_{10,11}=4$, $c_{11,12}=23$.

6) Чотирнадцять вершин з'єднані однонаправленими дугами, довжини яких складають: $c_{1,2}=10$, $c_{1,3}=2$, $c_{1,4}=7$, $c_{1,7}=22$, $c_{1,9}=33$, $c_{1,11}=15$, $c_{2,4}=59$, $c_{2,6}=25$, $c_{2,8}=40$, $c_{2,9}=50$, $c_{3,8}=35$, $c_{3,10}=12$, $c_{4,6}=2$, $c_{4,8}=13$, $c_{5,10}=56$, $c_{6,7}=51$, $c_{6,12}=37$, $c_{7,8}=7$, $c_{7,10}=21$, $c_{7,14}=34$, $c_{8,13}=11$, $c_{8,14}=60$, $c_{9,11}=71$, $c_{9,12}=32$, $c_{10,12}=66$, $c_{10,14}=24$, $c_{11,14}=43$, $c_{12,13}=17$, $c_{12,14}=16$, $c_{13,14}=56$.

4. Хід виконання роботи

- 1) уважно розібрати теоретичні відомості щодо задачі про найкоротшу путь;
- 2) разом з викладачем вибрати варіант завдання;
- 3) скласти алгоритм розв'язання задачі про найкоротшу путь використовуючи той засіб програмування, який на Вашу думку найбільш зручний для цієї задачі;
- 4) знайти розв'язок задачі про найкоротшу путь для декількох вершин-початків та вершин-кінців;
- 5) проаналізувати результати і зробити висновки щодо правильності отриманих розрахунків та реалізованого алгоритму;
- 6) оформити звіт.

5. Зміст звіту

- 1) завдання до роботи;
- 2) рисунок сітки до задачі;
- 3) лістинг програми;
- 4) результати розрахунків;
- 5) висновки.

6. Контрольні питання

- 1) У чому полягає задача про найкоротшу путь?
- 2) Які алгоритми Ви знаєте розв'язання задачі про найкоротшу путь?
- 3) Поясніть, як виконується пошук найкоротшої путі алгоритмом Мінті?
- 4) Поясніть, як виконується пошук найкоротшої путі алгоритмом Дейкстри?
- 5) Які труднощі виникли у Вас при реалізації алгоритму на ЕОМ? Як Ви їх подолали?
- 6) Чи піддається алгоритм, який Ви склали універсалізації?

7. Література

Ермольев Ю.М., Ляшко И.И., Михалевич В.С., Тюптя В.И., Математические методы исследования операций: Учебн. пособие для вузов. - К.: Вища школа, 1979. – 312с.

ЛАБОРАТОРНА РОБОТА 3

“Складання алгоритму розрахунку потоку максимальної величини”

Мета роботи: навчитися складати та реалізовувати алгоритм розрахунку потоку максимальної величини

1. Короткі теоретичні відомості з алгоритму розв’язання задачі про потік максимальної величини

Нехай дана множина вершин $I=\{1,\dots,n\}$ та множина дуг U , що зв’язують ці вершини. Для кожної дуги (i,j) задане число r_{ij} , яке означає пропускну спроможність дуги (i,j) . Вимірюватись число r_{ij} може, наприклад, в одиницях товару в день, або – у вагонах за годину, або – у кілобайтах за секунду. Це залежить від прикладного змісту задачі.

Із вершини 1 у вершину n дещо (одиниці товару, вагони або кілобайти інформації) потрібно перенести, використовуючи дуги із множини U . Кожній вершині i присвоєне значення інтенсивності d_i так, що вершина-джерело має інтенсивність $d_1=d$, а вершина-стік – інтенсивність $d_n=-d$. Всім іншим вершинам присвоюється інтенсивність $d_i=0$, $1<i<n$. Нульове значення інтенсивності означає, що кількість одиниць потоку, яка до вершини приходить, дорівнює кількості одиниць потоку, яка вершину залишає.

Потоком в сітці (I,U,r_{ij}) називається сукупність величин x_{ij} , для яких виконані співвідношення:

$$0 \leq x_{ij} \leq r_{ij}, (i, j) \in U \quad (3.1)$$

$$\begin{aligned} \sum_j x_{1j} &= d, \\ \sum_{j:(i,j) \in U} x_{ij} - \sum_{j:(j,i) \in U} x_{ji} &= 0, \quad 1 < i < n \\ - \sum_j x_{jn} &= -d \end{aligned} \quad (3.2)$$

де $\sum_{j:(i,j) \in U} x_{ij}$ – це величина потоку, що виходить із вершини i , $\sum_{j:(j,i) \in U} x_{ji}$ – величина потоку, який входить у вершину i .

Задача про потік максимальної величини полягає у тому, щоб знайти максимально можливе значення потоку d для заданої сітки (I,U,r_{ij}) : $d \rightarrow \max$.

Розіб’ємо множину вершин U на дві множини, що не перетинаються: R , що містить вершину-джерело потоку 1, і $\bar{R}$, що містить вершину-стік n . *Розрізом* сітки (I,J) називається множина дуг, початок яких належить множині R , а кінець – множині $\bar{R}$: $\{(i, j) \in U : i \in R, j \in \bar{R}\}$. *Пропускною спроможністю розрізу* $(R, \bar{R})$ називається величина:

$$C_R = \sum_{i \in R, j \in \bar{R}} r_{ij} \quad (3.3)$$

Покажемо, що величина потоку d не більше пропускну спроможності всякого розрізу:

$$d \leq C_R \quad (3.4)$$

Значення x_{ij} повинні задовольняти заданим умовам (3.1), (3.2). Просумуємо рівняння (3.2) для всіх вершин i , що належать множині R , і отримаємо:

$$\sum_{i \in R, j \in \bar{R}} x_{ij} - \sum_{i \in R, j \in R} x_{ij} = d \quad (3.5)$$

Оскільки кожне значення x_{ij} повинне бути меншим за значення пропускної спроможності r_{ij} , то перший доданок суми (3.5) має обмеження згори:

$$\sum_{i \in R, j \in \bar{R}} x_{ij} \leq \sum_{i \in R, j \in \bar{R}} r_{ij} = C_R. \quad (3.6)$$

Оскільки кожне значення x_{ij} повинне бути більшим за нульове значення, то другий доданок суми (3.5) також має обмеження згори:

$$- \sum_{i \in \bar{R}, j \in R} x_{ij} \leq 0. \quad (3.7)$$

Отже, вся сума (3.5) має обмеження згори:

$$d = \sum_{i \in R, j \in \bar{R}} x_{ij} - \sum_{i \in \bar{R}, j \in R} x_{ij} \leq C_R, \quad (3.8)$$

Таким чином, доведено, що умова (3.4) виконана. Оскільки ця умова виконана для будь-якого розрізу, то виконана також умова:

$$d \leq \min_R C_R \quad (3.9)$$

Значить, якщо може бути знайдений розріз мінімальної пропускної спроможності і знайдений потік x , для якого

$$d = \min_R C_R, \quad (3.10)$$

то це - потік максимальної величини. Як буде показано далі, такий розріз, дійсно, завжди може бути знайдений і тому справедливе наступне твердження:

$$\text{Величина потоку максимальної величини рівна пропускній спроможності мінімального розрізу} \quad (3.11)$$

Розв'язання задачі про потік максимальної величини і спосіб відшукування розрізу, який задовольняє умові (3.10) проілюструємо на прикладі.

Приклад 1. Дана сітка, що складається з шести вершин та дев'яти дуг і зображена на рисунку 3.1. В квадратику, приєднаному до дуги, указане значення пропускної спроможності цієї дуги. Потрібно знайти значення максимального потоку, яке можливе для даної сітки.

Спочатку розв'яжемо задачу *методом розрізів*. Для цього складемо список усіх можливих розрізів даної сітки. Достатньо вказати тільки вершини, що відносяться до множини R , яка містить вершину-джерело. Вершини, які залишились, відносяться до множини $\bar{R}$. Відшукаємо дуги, початок яких належить множині R , а кінець – множині $\bar{R}$, і підрахуємо значення пропускної спроможності за формулою (3.3).

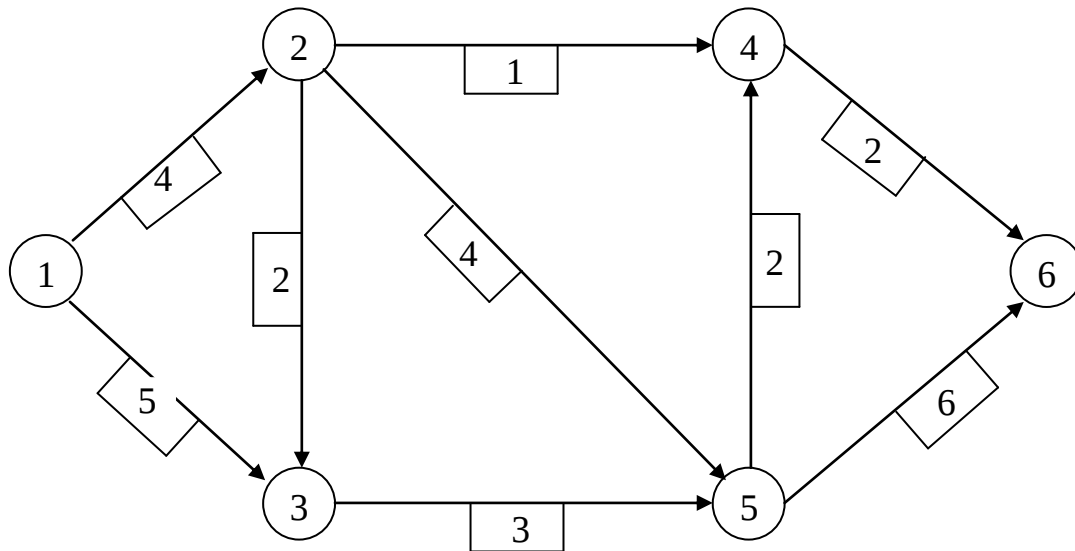


Рисунок 3.1. Сітка до прикладу 3.1

Визначимо найменше значення пропускної спроможності розрізу, яке за формулою (3.10) є значенням максимального потоку. Розрахунки наведені в таблиці 3.1, а ілюстрація до цих розрахунків представлена на рисунку 3.2.

Таблиця 3.1

Порядковий номер розрізу, i	Вершини, які відносяться до множини R	Розрахунок пропускної спроможності розрізу $(R, \bar{R})$	Значення пропускної спроможності розрізу, C_i
1	1	$r_{12}+r_{13}$	9
2	1, 2	$r_{24}+r_{25}+r_{23}+r_{13}$	12
3	1, 3	$r_{12}+r_{35}$	7
4	1, 4	$r_{12}+r_{13}+r_{46}$	11
5	1, 5	$r_{12}+r_{13}+r_{54}+r_{56}$	17
6	1, 2, 3	$r_{24}+r_{25}+r_{35}$	8
7	1, 2, 4	$r_{13}+r_{23}+r_{25}+r_{46}$	13
8	1, 2, 5	$r_{13}+r_{23}+r_{24}+r_{54}+r_{56}$	16
9	1, 3, 4	$r_{12}+r_{35}+r_{46}$	9
10	1, 3, 5	$r_{12}+r_{56}$	10
11	1, 4, 5	$r_{12}+r_{13}+r_{46}+r_{56}$	17
12	1, 2, 3, 4	$r_{35}+r_{25}+r_{46}$	9
13	1, 2, 3, 5	$r_{24}+r_{54}+r_{56}$	9
14	1, 2, 4, 5	$r_{13}+r_{23}+r_{46}+r_{56}$	17
15	1, 3, 4, 5	$r_{12}+r_{46}+r_{56}$	12
16	1, 2, 3, 4, 5	$r_{46}+r_{56}$	8
Максимальне значення потоку = мінімальне значення пропускної спроможності розрізу		$\max d = \min_i C_i = C_3 = 7$	

Мінімальним розрізом виявився розріз із значенням пропускної спроможності $C_2=7$ (виділений у таблиці і на рисунку сірим кольором). До мінімального розрізу відносяться дуги (1,2) та (3,5). Для того, щоб визначити поточні значення x_{ij} , які забезпечують потік максимальної величини $d=7$, присвоїмо дугам, що входять до мінімального розрізу, максимально можливі значення: $x_{12}=4$, $x_{35}=3$. Інші поточні значення присвоюємо таким чином, щоб були виконані умови (3.1), (3.2). Наприклад, так: $x_{13}=3$, $x_{24}=1$, $x_{25}=3$, $x_{23}=0$, $x_{46}=1$, $x_{56}=6$.

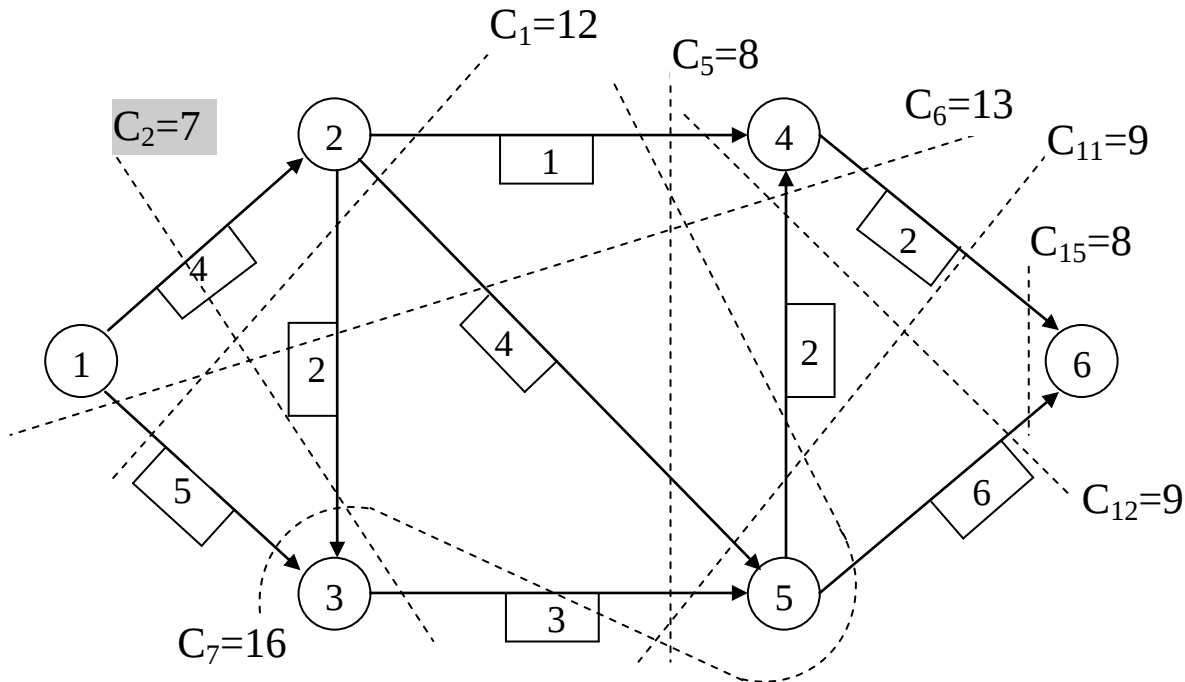


Рисунок 3.2. Розв'язання задачі про максимальний потік методом розрізів

Отже, **метод розрізів розв'язання задачі про потік максимальної величини** полягає у наступних кроках:

- Крок 0. Скласти список усіх можливих розрізів $(R, \bar{R})$ даної сітки.
- Крок 1. Для кожного розрізу знайти всі дуги, початок яких належить множині R , а кінець – множині $\bar{R}$, і розрахувати значення пропускної спроможності C_R .
- Крок 2. Максимальному значенню потоку присвоїти мінімальне значення пропускної спроможності.
- Крок 3. Визначити поточні значення потоку для кожної дуги, які реалізують потік максимальної величини.

Найскладнішим при реалізації методу розрізів на ЕОМ є визначення списку усіх можливих розрізів. Окрім того, при великій кількості вершин розрахунок усіх можливих розрізів приводить до тривалої часу спрацювання алгоритму.

Розв'яжемо тепер задачу із прикладу 3.1 *методом приєднаної сітки*. Задамо початковий потік. Найпростіше задати його нульовими значеннями: $x_{ij}=0$. Це не суперечить умовам (3.1), (3.2) і при реалізації алгоритму на ЕОМ не

вимагає побудови спеціальної процедури визначення початкового потоку. Поточне значення x_{ij} позначаємо на сітці уздовж дуги (i,j) так, як на рисунку 3.3. Величина потоку, який виходить із вершини-джерела, має значення $d=0$.

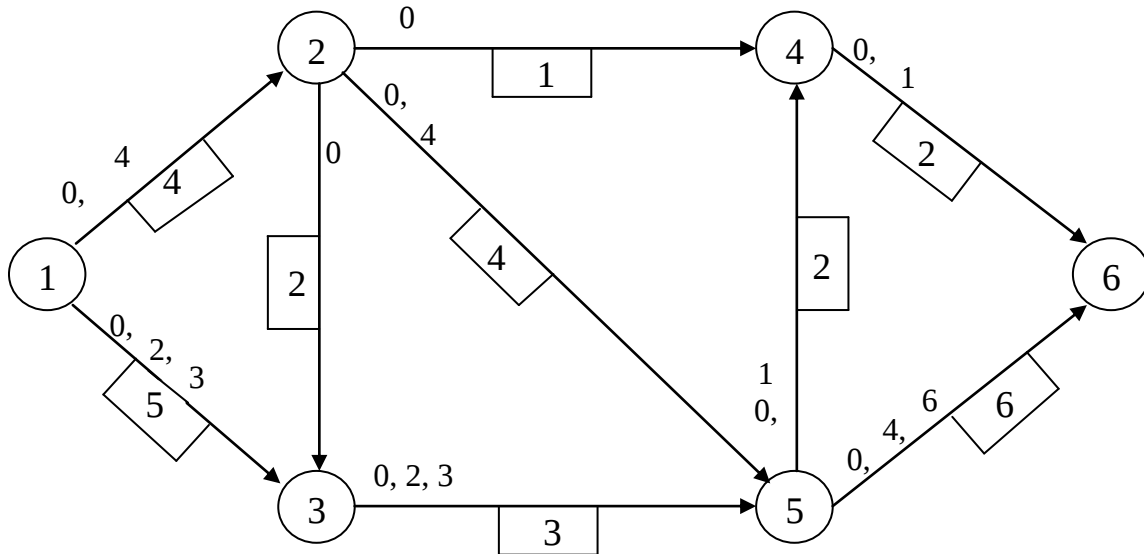


Рисунок 3.3. Розв'язання задачі про максимальний потік методом приєднаної сітки

Щоб поліпшити потік, побудуємо *приєднану* сітку (I, U') . Для цього кожній дузі із поточним значенням $x_{ij} < r_{ij}$ поставимо у відповідність *нормальну* дугу $(i, j) \in U'$ і присвоїмо їй значення $r'_{ij} = r_{ij} - x_{ij}$, а кожній дузі із поточним значенням $x_{ij} > 0$ поставимо у відповідність *зворотну* дугу $(j, i) \in U'$ і присвоїмо їй значення $r'_{ji} = x_{ij}$. Таким чином отримаємо сітку, яка представлена на рисунку 3.4, і відмічена як “дія 1”. Зауважимо, що величина r'_{ij} характеризує найбільшу кількість одиниць потоку, на яку може бути збільшено (уздовж нормальної дуги) або зменшено (уздовж зворотної дуги) поточне значення x_{ij} .

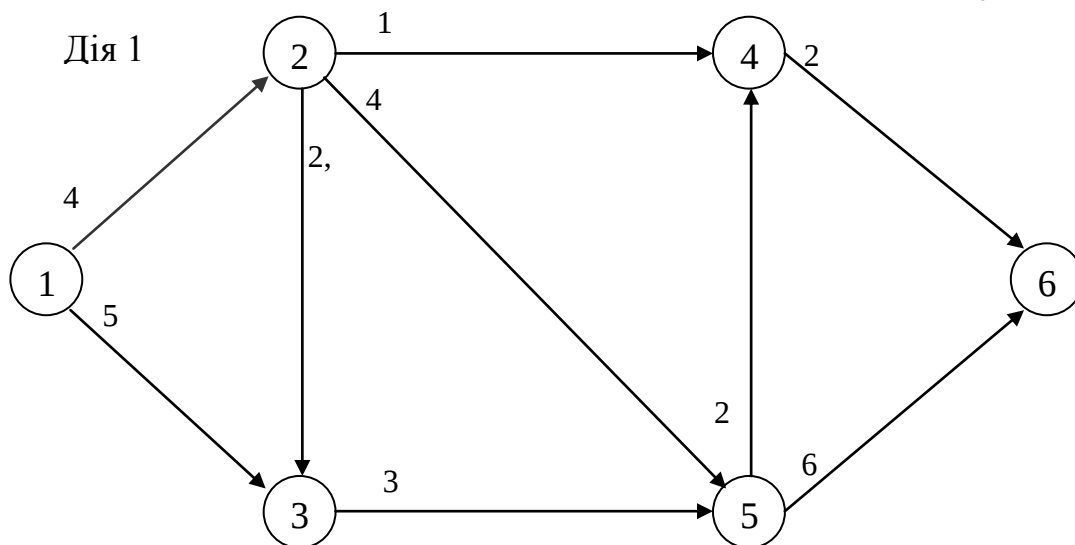


Рисунок 3.4. Розрахунок приєднаної сітки, дія 1

Відшукаємо в приєднаній сітці будь-який путь U'_p із 1 в 6. Наприклад, путь, що проходить через вершини 1, 2, 5, 6. Знайдемо найменше значення r'_{ij} уздовж цього путі:

$$\delta = \min_{(i,j) \in U'_p} r'_{ij}. \quad (3.12)$$

Перераховуємо поточні значення уздовж путі U'_p за формулами:

$$x_{ij} := \begin{cases} x_{ij} + \delta, & \text{if } (i, j) \in U'_p \text{ нормальна дуга} \\ x_{ij} - \delta, & \text{if } (i, j) \in U'_p \text{ зворотня дуга} \\ x_{ij}, & \text{if } (i, j) \notin U'_p \end{cases} \quad (3.13)$$

Зауважимо, що такий спосіб перерахунку забезпечує, по-перше, виконання умов (3.1), (3.2) і, по-друге, збільшення величини потоку d на величину δ :

$$d' = d + \delta. \quad (3.14)$$

Маємо $U'_p = \{(1,2);(2,5);(5,6)\}$, $\delta = \min\{4, 4, 6\} = 4$ і нові поточні значення:

$$x_{12} := x_{12} + \delta = 4$$

$$x_{25} := x_{25} + \delta = 4,$$

$$x_{56} := x_{56} + \delta = 4$$

які на рисунку 3.3 зображені через кому після попереднього значення уздовж своєї дуги. Поліпшене значення потоку складає $d=4$.

Змінювання поточних значень x_{ij} означає змінювання значень r'_{ij} приєднаної сітки. Змінена приєднана сітка представлена на рисунку 3.5 і відмічена як “дія 2”. Зверніть увагу на те, що деякі дуги при перерахунку зникають – на рисунку вони відмічені сірим кольором. Нульове значення на цих дугах означає, що вони не можуть бути використані для поліпшення потоку.

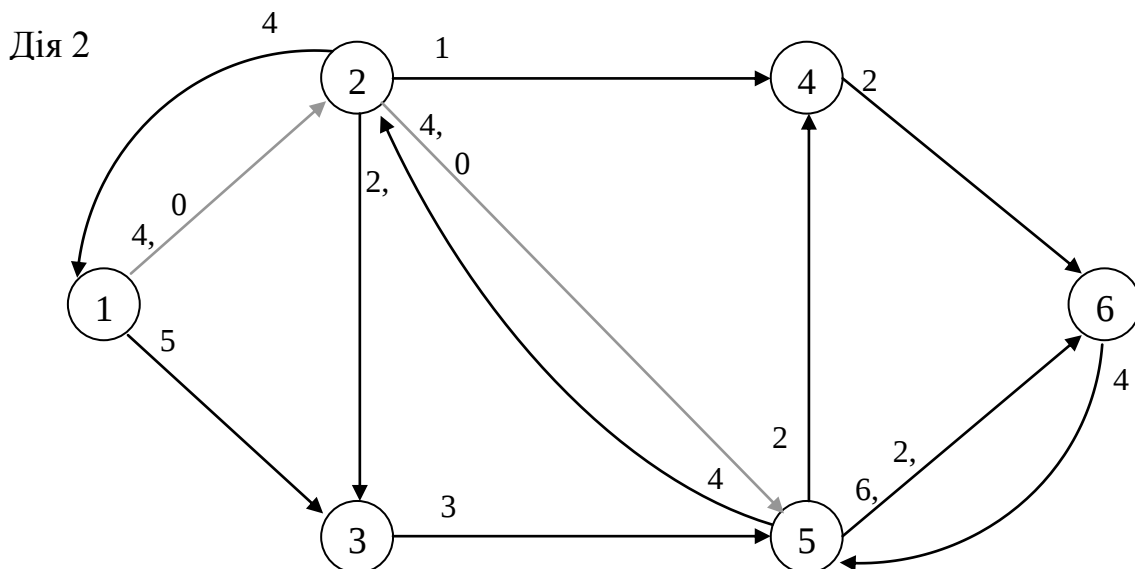


Рисунок 3.5. Розрахунок приєднаної сітки, дія 2

Пошукаємо путь в приєднаній сітці від 1 до 6 (дуги, помічені сірим кольором використовувати не можна). Оберемо, наприклад, путь, який

проходить через вершини 1,3,5,6. Маємо $U'_p = \{(1,3);(3,5);(5,6)\}$, $\delta = \min\{5, 3, 2\} = 2$ і нові поточні значення:

$$x_{13} := x_{13} + \delta = 2$$

$$x_{35} := x_{35} + \delta = 2.$$

$$x_{56} := x_{56} + \delta = 6$$

Поліпшене значення потоку складає $d=4+2=6$.

Знову побудуємо приєднану сітку до нової сітки (рис. 3.6) і пошукаємо путь від 1 до 6.

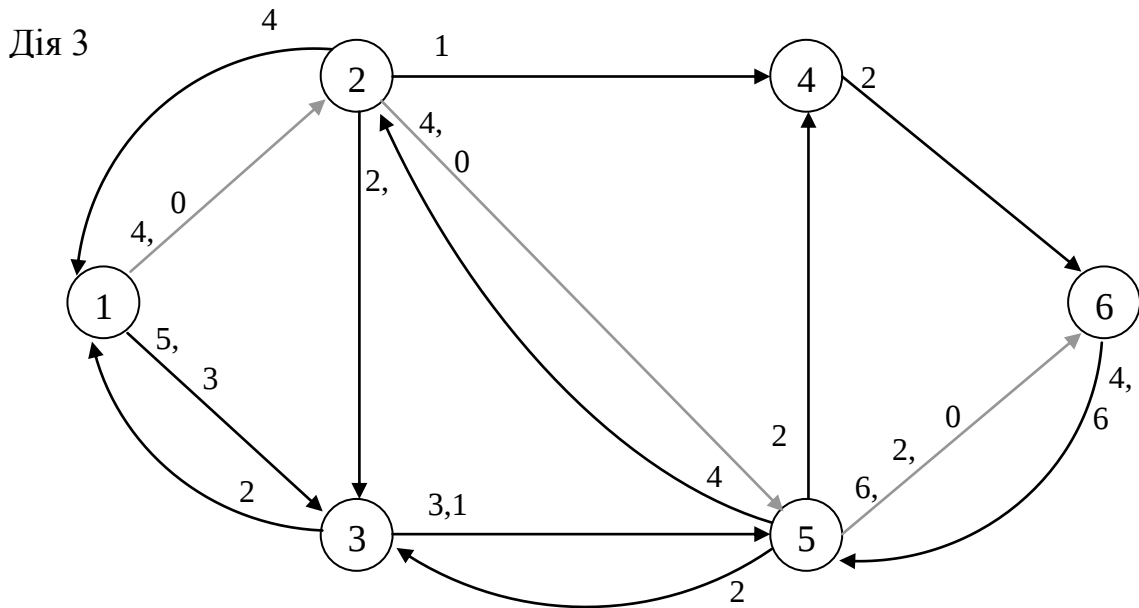


Рисунок 3.6. Розрахунок приєднаної сітки, дія 3

Оберемо, наприклад, путь, який проходить через вершини 1,3,5,4,6. Маємо $U'_p = \{(1,3);(3,5);(5,4);(4,6)\}$, $\delta = \min\{3, 1, 2, 2\} = 1$ і нові поточні значення:

$$x_{13} := x_{13} + \delta = 3$$

$$x_{35} := x_{35} + \delta = 3$$

$$x_{54} := x_{54} + \delta = 1$$

$$x_{46} := x_{46} + \delta = 1$$

Поліпшене значення потоку складає $d=6+1=7$. Знову побудуємо приєднану сітку (рис. 3.7) і пошукаємо путь від 1 до 6. Очевидно, що жодного путі від 1 до 6 не існує. Це означає, що подальше поліпшення потоку не існує. Досягнуте максимальне значення потоку складає $d=7$ і співпадає із отриманим раніше значенням методом розрізів. Поточні значення x_{ij} відрізняються від отриманих вище, оскільки значення потоку $d=7$ може бути реалізоване багатьма способами.

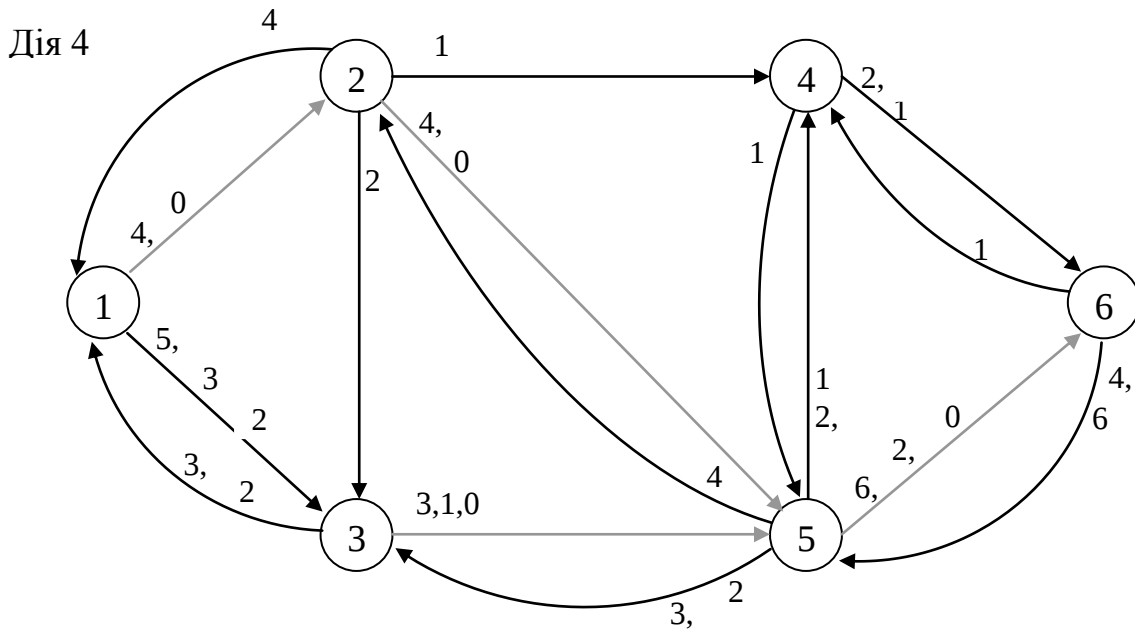


Рисунок 3.7. Розрахунок приєднаної сітки, дія 4

Доведемо справедливості наступного твердження:

Потік у сітці максимальний тоді і тільки тоді, коли приєднана сітка не містить жодного путі з вершини-джерела до вершини-стоку.

Дійсно, нехай R - безліч усіх вершин сітки, які є досяжними із вершини 1 в приєднаній сітці. Тобто не має дуг (ні нормальних, ні зворотних), що поєднують вершину із множини R із будь-якою іншою вершиною сітки. Оскільки немає жодної нормальної дуги із R в $\bar{R}$, то для всіх дуг сітки $(i, j) \in U : i \in R, j \in \bar{R}$ повинно досягатись максимально можливе поточне значення $x_{ij}=r_{ij}$. Оскільки немає жодної зворотної дуги із R в $\bar{R}$, то для всіх дуг $(i, j) \in U : i \in \bar{R}, j \in R$ повинно досягатись мінімально можливе поточне значення $x_{ij}=0$. На основі рівності (3.5) і визначення (3.3) маємо:

$$d = \sum_{\substack{i \in R \\ j \in \bar{R}}} x_{ij} - \sum_{\substack{i \in \bar{R} \\ j \in R}} x_{ij} = \sum_{\substack{i \in R \\ j \in \bar{R}}} r_{ij} - \sum_{\substack{i \in \bar{R} \\ j \in R}} 0 = C_R. \quad (3.15)$$

Рівність (3.15) означає, що знайдений потік, який задовольняє умові (3.10), а значить знайдений потік максимальної величини. Розріз мінімальної пропускної спроможності C_R відповідає розподіленню вершин сітки на такі R і $\bar{R}$, що R - множина всіх вершин, досяжних із вершини-джерела в приєднаній сітці. На рисунку 3.7 досяжними з вершини 1 в приєднаній сітці є тільки вершина 3, значить множина $R = \{1, 3\}$. Пропускна спроможність цього розрізу $C_R = r_{12} + r_{35} = 4 + 3 = 7$.

Зауваження. Якщо пропускні спроможності дуг раціональні, потрібно спочатку привести їх до цілочисельних, і тільки після цього приступати до розрахунків задачі.

Алгоритм розв'язання задачі про потік максимальної величини методом приєднаної сітки полягає у наступних кроках:

Крок 0. Побудувати початковий потік, наприклад нульовий. Перейти до кроку 1.

Крок 1. Побудувати приєднану сітку за формулами:

$$r'_{ij} = \begin{cases} r_{ij} - x_{ij}, & \text{if } x_{ij} < r_{ij}, \quad (i, j) \in U \\ x_{ji}, & \text{if } x_{ji} > 0, \quad (j, i) \in U \end{cases}. \quad (3.16)$$

Крок 2. Знайти путь U'_p в приєднаній сітці із вершини 1 у вершину n . Якщо жодного путі не знайдено, то потік x_{ij} є потоком максимальної величини $d = \sum_j x_{1j}$ і кінець алгоритму. Інакше

перейти до кроку 3.

Крок 3. Перерахувати поточні значення за формулами:

$$x_{ij} := \begin{cases} x_{ij} + \delta, & \text{if } (i, j) \in U'_p, \quad (i, j) \in U \\ x_{ij} - \delta, & \text{if } (i, j) \in U'_p, \quad (j, i) \in U, \text{ де } \delta = \min_{(i,j) \in U'_p} r'_{ij}. \\ x_{ij}, & \text{if } (i, j) \notin U'_p \end{cases} \quad (3.17)$$

Перейти до кроку 1.

Метод приєднаної сітки зручний при реалізації на ЕОМ і дуже швидко приводить до розв'язку задачі.

2. Завдання

Для даного варіанту сітки:

- 1) скласти та нарисувати сіткову модель;
- 2) скласти та реалізувати на ЕОМ алгоритм відшукування потоку максимальної величини;
- 3) перевірити складений алгоритм на простій моделі, для якої значення максимального потоку відоме;
- 4) зробити розрахунки за даними варіанту завдання.

При складанні алгоритму передбачити можливість зміни пропускної спроможності будь-якої дуги.

3. Варіанти

1)

дуга, (i,j)	1,2	1,3	1,4	1,5	2,3	2,6	2,8	2,10	3,4	3,7	4,5	
пропускна спроможність дуги, $r_{i,j}$	12	63	4	5	1	32	13	3	5	4	21	
дуга (i,j)	4,8	4,9	5,6	5,7	5,8	6,7	6,9	7,8	7,9	7,10	8,9	9,10
пропускна спроможність дуги, $r_{i,j}$	23	12	4	5	16	21	25	28	29	31	15	2

2)

дуга, (i,j)	1,3	1,6	1,7	1,9	2,3	2,4	2,6	2,7	3,5	3,6	3,8
пропускна спроможність дуги, $\Gamma_{i,j}$	31	35	21	64	51	25	22	3	14	8	19
дуга, (i,j)	3,11	4,5	4,6	4,8	4,10	5,6	5,8	5,9	6,8	6,10	7,9
пропускна спроможність дуги, $\Gamma_{i,j}$	25	16	10	8	11	22	25	13	6	3	12
дуга, (i,j)	7,11	8,10	9,10	9,11							
пропускна спроможність дуги, $\Gamma_{i,j}$	11	15	25	16							

3)

дуга, (i,j)	1,2	1,5	1,7	1,9	2,3	2,5	2,7	2,9	3,10	3,11	4,5
пропускна спроможність дуги, $\Gamma_{i,j}$	25	28	19	31	16	2	3	12	62	31	15
дуга, (i,j)	4,6	4,8	4,9	5,7	5,8	5,11	6,7	6,9	6,10	7,8	7,9
пропускна спроможність дуги, $\Gamma_{i,j}$	4	2	13	11	25	28	19	2	5	6	3
дуга, (i,j)	7,10	8,9	8,10	9,10	9,11						
пропускна спроможність дуги, $\Gamma_{i,j}$	1	12	25	29	3						

4)

дуга, (i,j)	1,2	1,3	1,4	1,5	1,7	2,4	2,5	2,7	2,8	2,9	2,11
пропускна спроможність дуги, $r_{i,j}$	21	15	6	4	28	9	1	2	13	27	45
дуга, (i,j)	3,4	3,5	3,7	3,8	4,6	4,9	4,10	5,6	5,8	5,9	6,7
пропускна спроможність дуги, $r_{i,j}$	21	4	32	35	8	14	21	13	3	1	14
дуга, (i,j)	6,8	6,9	7,9	7,11	8,9	8,10	9,10	9,11	10,11		
пропускна спроможність дуги, $r_{i,j}$	18	25	29	31	18	29	31	35	38		

4. Хід виконання роботи

- 1) уважно розібрати теоретичні відомості щодо задачі про потік максимальної величини;
- 2) разом з викладачем вибрати варіант завдання;
- 3) скласти алгоритм розв'язання задачі про потік максимальної величини використовуючи той засіб програмування, який на Вашу думку найбільш зручний для цієї задачі, та реалізувати його на ЕОМ;
- 4) виконати перевірку алгоритму;
- 5) виконати розрахунок потоку максимальної величини за даними Вашого варіанту завдання;
- 6) проаналізувати результати і зробити висновки щодо правильності отриманих розрахунків та реалізованого алгоритму;
- 7) оформити звіт.

5. Зміст звіту

- 1) завдання до роботи;
- 2) рисунок сітки до задачі;
- 3) лістинг програми з коментарями;
- 4) результати розрахунків;
- 5) висновки.

6. Контрольні питання

- 1) У чому полягає задача про потік максимальної величини? Дайте математичну постановку задачі.
- 2) Які методи розв'язання задачі про потік максимальної величини Ви знаєте?
- 3) Що є розрізом сітки?

- 4) Як знаходиться потік максимальної величини методом розрізів?
- 5) Поясніть алгоритм відшукування потоку максимальної величини методом приєднаної сітки.
- 6) Як побудувати приєднану сітку?
- 7) Яка дуга нормальна, а яка зворотна в приєднаній сітці?
- 8) Поясніть алгоритм відшукування потоку максимальної величини, який склали Ви.
- 9) Які труднощі виникли у Вас при реалізації алгоритму на ЕОМ?

7. Література

Математическое моделирование, Ред. Дж.Эндрюс, Мак-Лоун Р. и др., М., 1979.

ЛАБОРАТОРНА РОБОТА 4

“Розв’язання задачі про оптимальний потік”

Мета роботи: навчитися складати та реалізовувати алгоритм розв’язання задачі про оптимальний потік.

1. Короткі теоретичні відомості з алгоритму розв’язання задачі про оптимальний потік

В графі (I, U) кожній вершині поставлено у відповідність число d_i - інтенсивність вершини. Вершина є джерелом, якщо $d_i > 0$, стоком, якщо $d_i < 0$, і нейтральна, якщо $d_i = 0$. *Потоком* в отриманій сітці називається сукупність величин x_{ij} , $(i, j) \in U$, які задовольняють співвідношенням:

$$0 \leq x_{ij} \leq r_{ij} \quad (4.1)$$

$$\sum_j x_{ij} - \sum_j x_{ji} = d_i, \quad (4.2)$$

де $\sum_j x_{ij}$ - величина потоку, що виходить із вершини i , $\sum_j x_{ji}$ - величина потоку, який входить у вершину i , r_{ij} - пропускна спроможність дуги (i, j) . Для кожної дуги задана *вартість* використання цієї дуги c_{ij} у розрахунку на одиницю потоку. Поняття вартості в цій задачі умовне і може оцінюватись як у гривнях, так і у кілометрах, у годинах, у кілобайтах за годину і т.і. в залежності від прикладного змісту задачі.

Задача про оптимальний потік полягає у відшуванні такого потоку у сітці, який мінімізує значення величини $\sum_{(i,j) \in U} c_{ij} \cdot x_{ij}$ при обмеженнях (4.1), (4.2).

Для розв’язання задачі про оптимальний потік користуються методом потенціалів. Сформулюємо **критерій оптимальності потоку**:

Для того, щоб потік x_{ij} був оптимальним, необхідно і достатньо існування таких величин v_i , названих *потенціалами* вершин, для яких виконані співвідношення

$$v_j - v_i \leq c_{ij}, \quad \text{if } x_{ij} = 0 \quad (4.3)$$

$$v_j - v_i = c_{ij}, \quad \text{if } 0 < x_{ij} < r_{ij} \quad (4.4)$$

$$v_j - v_i \geq c_{ij}, \quad \text{if } x_{ij} = r_{ij} \quad (4.5)$$

Зверніть увагу, що різниця потенціалів менша, рівна або більша вартості дуги в залежності від потокового значення:

- якщо потокове значення нульове (мінімальне), то різниця потенціалів повинна бути меншою за вартість дуги;
- якщо потокове значення максимальне, то різниця потенціалів повинна бути більшою за вартість дуги;
- інакше різниця потенціалів повинна бути рівною вартості дуги.

Метод потенціалів відшукування оптимального потоку в сітці полягає в наступному. За умовами (4.1) та (4.2) будується початковий потік. Далі виконується пошук потенціалів v_i за рівностями (4.4). Після цього перевіряється оптимальність потоку за критерієм оптимальності. Якщо потік не оптимальний, то будується поліпшений потік і т. д.

Нехай заданий початковий потік x . Граф (I, U^x) такий, що множина його дуг $U^x = \{(i, j): 0 < x_{ij} < r_{ij}\}$, називається *опорою* потоку x . Якщо опора U^x є зв'язаним графом, то потенціали вершин визначаються без ускладнень і тому такий випадок називають *невиродженням*.

Нехай потік x є неvirодженням. Потенціали v_j визначаються так. Для деякої вершини i приймаємо $v_i = 0$. Далі для всіх j таких, що $(i, j) \in U^x$ або $(j, i) \in U^x$ визначаємо потенціал v_j так, щоб була виконана рівність (4.4):

$$v_j = c_{ij} + v_i, \quad \text{if } (i, j) \in U^x \quad (4.6)$$

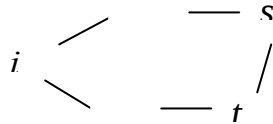
$$v_j = v_i - c_{ij}, \quad \text{if } (j, i) \in U^x$$

Визначивши потенціал вершини j , визначаємо потенціали вершини, зв'язаних з вершиною j і т. д. Оскільки U^x зв'язаний граф, то таким чином визначимо потенціали всіх вершин v_i . Послідовність визначення потенціалів запам'ятовується, оскільки нею користуються для побудови поліпшеного потоку.

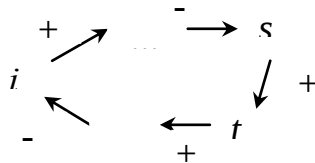
Якщо потік x вироджений, то U^x незв'язаний граф, і послідовність визначення потенціалів закінчиться раніше, ніж обійде всі вершини сітки (I, U) . Теоретично слід розглянути розширену задачу: у кожній компоненті зв'язності опори U^x вибрати вершину-джерело і зменшити її інтенсивність на малу величину ε . Після цього з'єднати її з фіктивної вершиною, причому фіктивній вершині приписати інтенсивність рівну $\varepsilon \cdot \kappa$, де κ - кількість компонент зв'язності опори U^x , а кожній фіктивній дузі - нульову вартість і необмежену пропускну спроможність. Практично для усунення виродження достатньо з'єднати компоненти зв'язності опори U^x потоком нульової величини.

Знайдені величини v_i використовуються для перевірки критерію оптимальності. Виконання умов (4.4), (4.5), (4.6) для всіх дуг сітки означає, що потік є оптимальним. В противному випадку виконується поліпшення потоку.

Опишемо, як побудувати поліпшений потік. Нехай не виконана умова (4.3) критерію оптимальності, тобто для дуги (s,t) значення потоку $x_{st}=0$, а різниця потенціалів більша за вартість дуги $v_t - v_s > c_{st}$. Відтворимо послідовність, за якою був визначений потенціал v_s : $i \dots s$. Відтворимо також послідовність, за якою був визначений потенціал v_t : $i \dots t$. Створимо з двох послідовностей цикл, з'єднавши s та t :



На дузі (s,t) маємо мінімальне (нульове) потокове значення. Виходячи з цього робимо висновок, що поліпшення потоку може бути тільки в сторону збільшення значення x_{st} . Тому позначаємо зв'язок (s,t) знаком “+” і обираємо напрямок в циклі уздовж дуги (s,t) . Помічаємо знаком “+” також дуги циклу, напрямок яких співпадає з напрямком в сітці, і знаком “-” дуги циклу, напрямок яких протилежний до напрямку в сітці:



Отриманий цикл називають *циклом перерахунку*. Позначимо множину дуг, що входить до циклу перерахунку U_P . На дугах із позначкою “+” при перерахунку потоку здійснюється збільшення потокового значення, а на дугах із позначкою “-” – зменшення. Позначимо підмножину дуг із позначкою “+” - U_P^+ , а підмножину дуг із позначкою “-” - U_P^- . Максимально можливе число δ , на яке може бути змінений потік x розраховується за формулою:

$$\delta = \min \{ \min_{U_P^+} x_{ij}, \min_{U_P^-} (r_{ij} - x_{ij}) \}. \quad (4.7)$$

Перерахунок потоку здійснюється за формулами:

$$x'_{ij} = \begin{cases} x_{ij} + \delta, & \text{if } (i, j) \in U_P^+ \\ x_{ij} - \delta, & \text{if } (i, j) \in U_P^- \\ x_{ij}, & \text{if } (i, j) \notin U_P \end{cases} \quad (4.8)$$

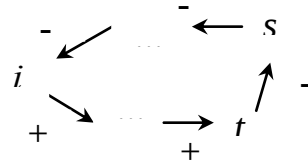
Покажемо, що потік x'_{ij} ліпше потоку x_{ij} . Дійсно,

$$\begin{aligned} \sum_{(i,j) \in U} c_{ij} x_{ij} - \sum_{(i,j) \in U} c_{ij} x'_{ij} &= \sum_{(i,j) \in U_P} c_{ij} x_{ij} - \sum_{(i,j) \in U_P^+} c_{ij} (x_{ij} + \delta) - \sum_{(i,j) \in U_P^-} c_{ij} (x_{ij} - \delta) = \\ &= \delta \cdot \left(\sum_{U_P^-} c_{ij} - \sum_{U_P^+} c_{ij} \right) = \delta \cdot (v_t - v_s - c_{st}) > 0 \end{aligned} \quad (4.9)$$

оскільки $v_t - v_s > c_{st}$.

Таким чином, перерахунок потоку за формулами (4.7), (4.8) гарантує, по-перше, виконання умов (4.1), (4.2) і, по-друге, зменшення величини потоку.

Нехай не виконана умова (4.5) критерію оптимальності, тобто для дуги (s,t) значення потоку максимальне $x_{st}=r_{st}$, а різниця потенціалів менша за вартість дуги $v_t-v_s < c_{st}$. Поліпшення потоку в цьому випадку може бути тільки в сторону зменшення значення x_{st} . Тому позначаємо зв'язок (s,t) знаком “-” і обираємо напрямок в циклі проти напрямку дуги (s,t) :



Знаходимо число δ у відповідності з (4.7) і визначаємо поліпшений потік за формулами (4.8). Тепер у доведенні (4.9) маємо:

$$\sum c_{ij}x_{ij} - \sum c_{ij}x'_{ij} = \delta \cdot (v_s - v_t + c_{st}) > 0,$$

оскільки $v_t - v_s < c_{st}$.

Нехай тепер не виконана умова (4.4) критерію оптимальності, тобто для дуги (s,t) значення потоку ні мінімальні ні максимальне, а різниця потенціалів не рівна за вартості дуги $v_t - v_s \neq c_{st}$. Якщо $v_t - v_s < c_{st}$, то напрямком в циклі U_p слід вибирати за напрямком дуги (s,t) , а якщо $v_t - v_s > c_{st}$, то напрямком в циклі U_u слід вибирати проти напрямку дуги (s,t) . Тоді потік (4.8) буде ліпше попереднього.

Розглянемо алгоритм розв'язання задачі про оптимальний потік на прикладі.

Приклад 1. П'ять вершин $I=\{1,2,3,4,5\}$ з'єднані дугами $U=\{12,23,31,42,43,45,53\}$. Задані значення інтенсивностей вершин $d_1=7, d_2=-3, d_3=-10, d_4=8, d_5=-2$. Пропускна спроможність всіх дуг однакова і складає $r_{ij}=5$. Вартості використання дуг задані наступними значеннями: $c_{12}=20, c_{23}=40, c_{31}=30, c_{42}=10, c_{43}=50, c_{45}=50, c_{53}=10$. Потрібно знайти значення потоку x , який мінімізує значення величини $\sum_{(i,j) \in U} c_{ij} \cdot x_{ij}$.

Розв'язання. За умовами задачі складемо сітку (рис. 4.1). Побудуємо початковий потік так, щоб виконувались умови (4.1), (4.2). Розподілимо потік із вершини 1: $x_{12}=4, x_{13}=3$. Розподілимо потік із вершини 4: $x_{42}=0, x_{43}=3, x_{45}=5$. Перепишемо умову (4.2) для вершини 2:

$$x_{23} - x_{12} - x_{42} = -3.$$

Звідси знаходимо значення потоку для дуги (2,3):

$$x_{23} = x_{12} + x_{42} - 3 = 1.$$

Аналогічно знаходимо значення потоку для дуги (5,3):

$$x_{53} = x_{45} - 2 = 3.$$

Перевіряємо виконання умови (4.2) для вершини 3:

$$-x_{13} - x_{23} - x_{43} - x_{53} = -10.$$

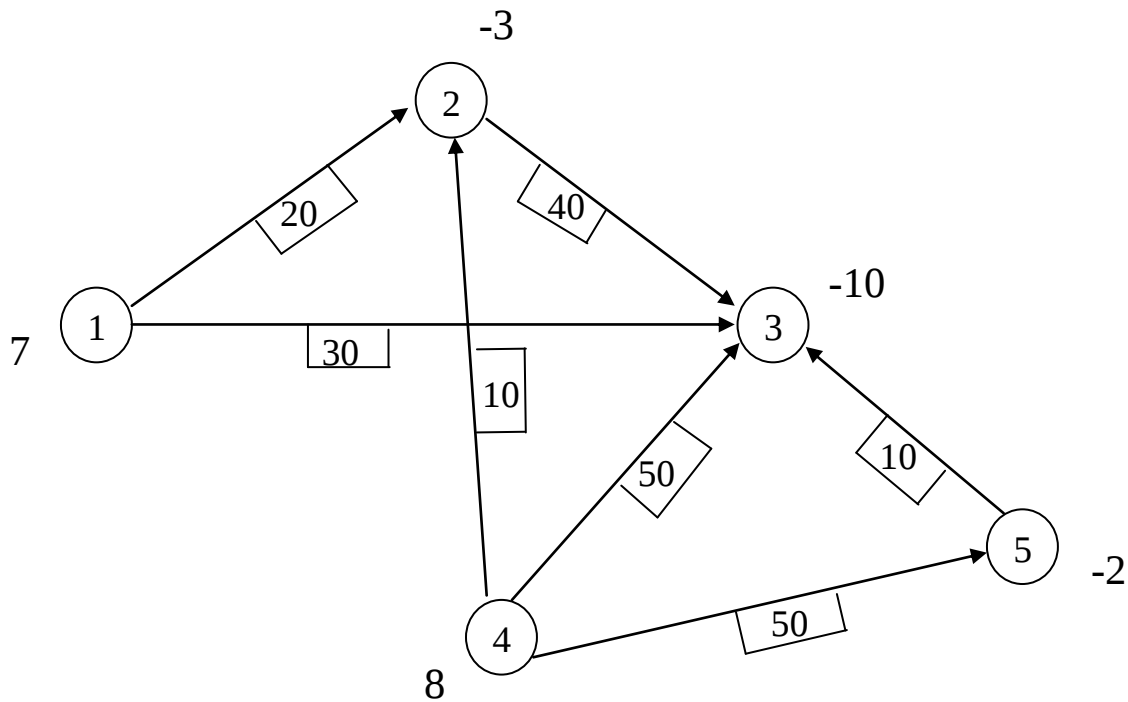


Рисунок 4.1 Сітка до прикладу 1

Отже, початковий потік визначений (рис. 4.2).

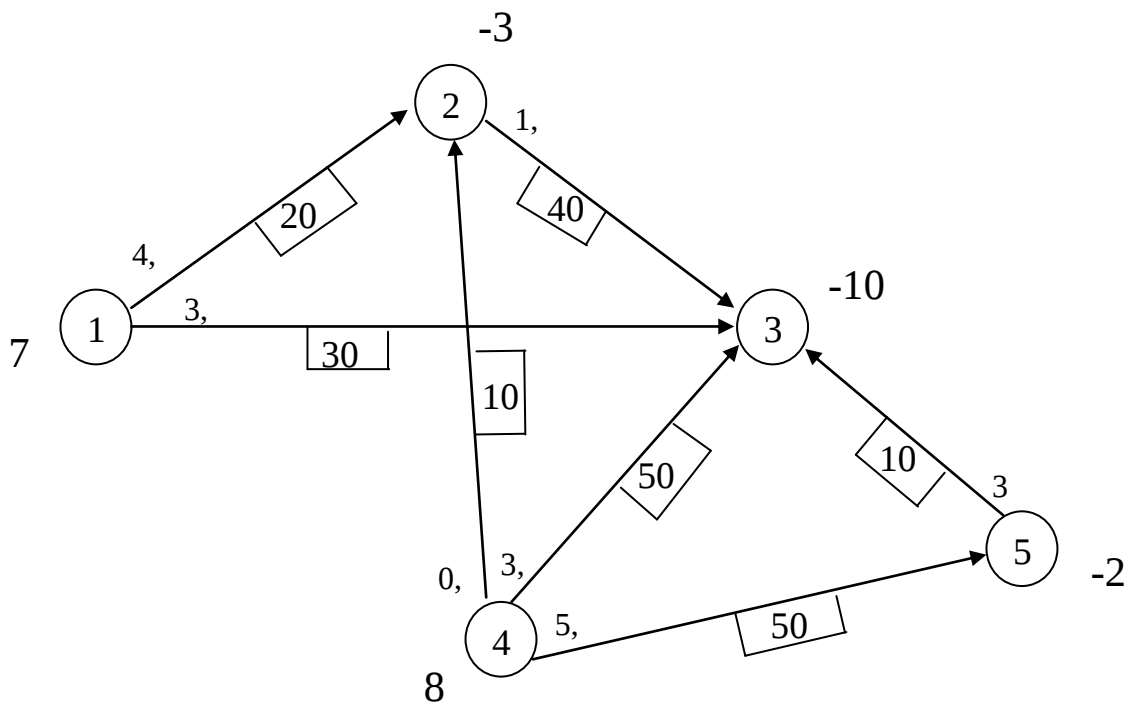


Рисунок 4.2 Визначення початкового потоку

Опора цього потоку $U^x = \{12, 13, 23, 43, 53\}$ є зв'язаний граф. Визначаємо потенціали вершин. Нехай $v_1 = 0$. З вершиною 1 зв'язані вершини 2 та 3. З умови (4.4) маємо:

$$v_2 - v_1 = 20 \Rightarrow v_2 = 20,$$

$$v_3 - v_1 = 30 \Rightarrow v_3 = 30.$$

Із відомого тепер значення потенціалу вершини 3 за умовою (4.4) знаходимо потенціали вершин 4 та 5:

$$v_3 - v_4 = 50 \Rightarrow v_4 = -20,$$

$$v_3 - v_5 = 10 \Rightarrow v_5 = 20.$$

Відшукування потенціалів таким способом представлено формулою (4.6). Послідовність визначення потенціалів важлива для побудови циклу перерахунку, тому розрахунок потенціалів зручно представляти у вигляді, представленому на рисунку 4.3.

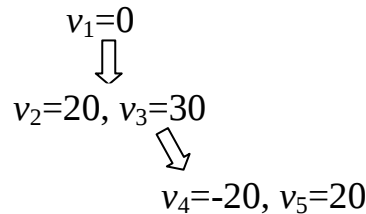


Рисунок 4.3. Послідовність розрахунку потенціалів вершин

Перевіримо оптимальність потоку за критерієм. Відшукуємо дуги, для яких $x_{ij}=0$, і перевіряємо умову (4.3):

$$v_2 - v_4 = 20 + 20 > 10 \text{ не виконана умова (4.3)}$$

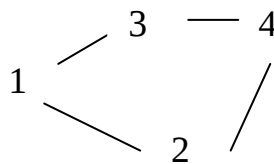
Відшукуємо дуги, для яких $x_{ij}=5$, тобто максимально можливе значення, і перевіряємо умову (4.5):

$$v_5 - v_4 = 20 + 20 < 50 \text{ не виконана умова (4.5)}$$

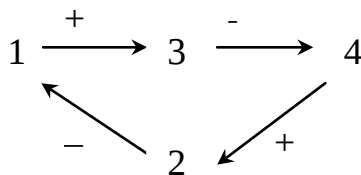
Для всіх інших дуг перевіряємо умову (4.4):

$$v_3 - v_2 = 30 - 20 \neq 40 \text{ не виконана умова (4.4)}$$

Отже, потік не оптимальний. Виберемо для поліпшення потоку дугу (4,2). Побудуємо цикл перерахунку U_P . З рисунку 4.3 відтворимо послідовність визначення потенціалу v_4 : $1 \rightarrow 3 \rightarrow 4$. Послідовність визначення потенціалу v_2 : $1 \rightarrow 2$ (див. рис. 4.3). З'єднуємо 4 та 2 і отримуємо наступний цикл:



Оскільки на дузі (4,2) маємо нульове потокове значення, то обираємо в циклі напрямок уздовж дуги (4,2). Помічаємо знаком “+” дуги циклу, напрямки яких співпадає з напрямком в сітці, і знаком “-” дуги циклу, напрямки яких протилежний до напрямку в сітці. Таким чином, отримуємо наступний цикл



перерахунку U_P :

Визначаємо максимально можливе число δ , на яке може бути змінений потік x за формулою (4.7):

$$\delta = \min \{ \min \{ x_{12}, x_{34} \}, \min \{ 5 - x_{13}, 5 - x_{42} \} \} = 2.$$

Перераховуємо потік x за формулами (4.8):

$$x'_{13} = x_{13} + \delta = 3 + 2 = 5$$

$$x'_{42} = x_{42} + \delta = 0 + 2 = 2$$

$$x'_{12} = x_{12} - \delta = 4 - 2 = 2$$

$$x'_{43} = x_{43} - \delta = 3 - 2 = 1$$

$$x'_{ij} = x_{ij} \text{ if } (i, j) \notin U_p$$

Помічаємо нові значення потоку на рисунку 4.4 уздовж дуги через кому після попереднього значення.

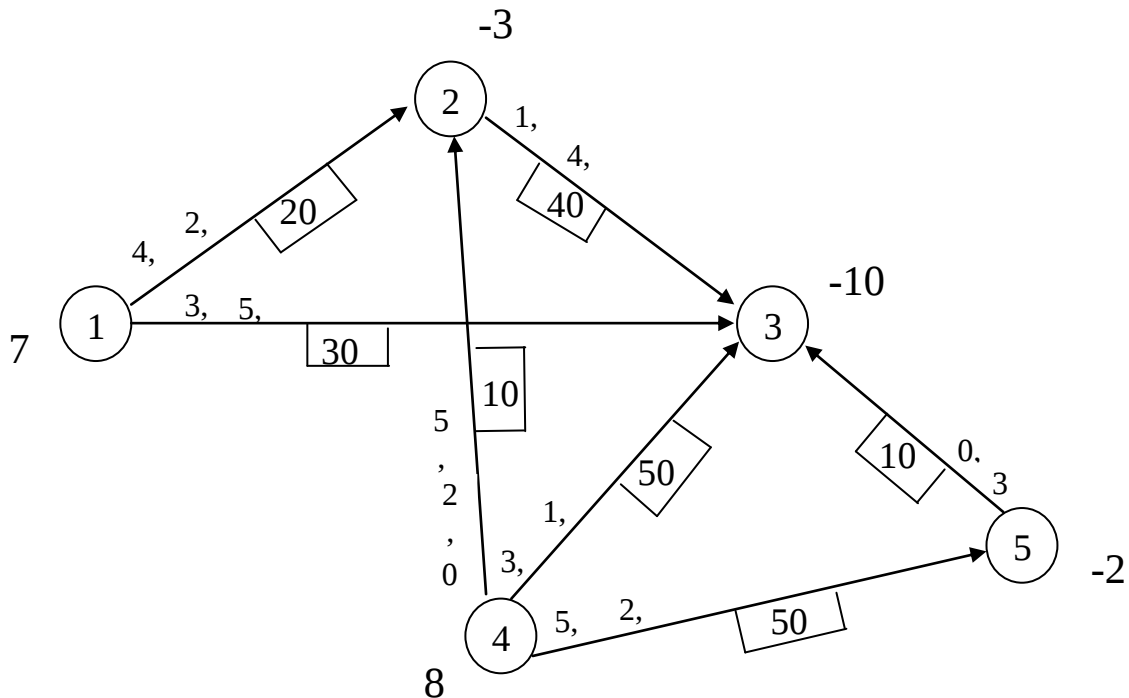


Рисунок 4.4. Розв'язання задачі про оптимальний потік

Перевіримо, чи дійсно потік поліпшився. Попереднє значення потоку складає

$$\sum_{(i,j) \in U} c_{ij} x_{ij} = 4 \cdot 20 + 3 \cdot 30 + 1 \cdot 40 + 0 \cdot 10 + 3 \cdot 50 + 5 \cdot 50 + 3 \cdot 10 = 640.$$

Діюче значення потоку складає

$$\sum_{(i,j) \in U} c_{ij} x'_{ij} = 2 \cdot 20 + 5 \cdot 30 + 1 \cdot 40 + 2 \cdot 10 + 1 \cdot 50 + 5 \cdot 50 + 3 \cdot 10 = 580.$$

Значення потоку зменшилось, значить перерахунок виконаний вірно.

Зауваження. Перевірку значення потоку обов'язково потрібно виконувати при реалізації алгоритму на ЕОМ. Якщо при перевірці виявилось, що значення потоку збільшилось, це означає наявність помилки або в побудові циклу перерахунку, або у виконанні перерахунку.

Перевіримо оптимальність нового потоку. Опора потоку $U^x = \{12, 23, 42, 43, 53\}$ є зв'язний граф. Знову визначаємо потенціали:

$$\begin{aligned}
 v_1 &= 0 \\
 \Downarrow \\
 v_2 &= 20 \\
 \Downarrow \\
 v_3 &= 60, v_4 = 10 \\
 \Downarrow \\
 v_5 &= 50
 \end{aligned}$$

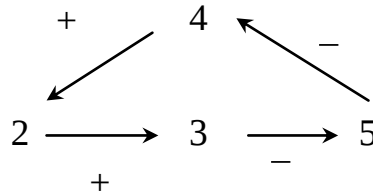
Перевіряємо оптимальність:

$$v_3 - v_1 = 60 > 30 \text{ виконана умова (4.5)}$$

$$v_5 - v_4 = 40 < 50 \text{ не виконана умова (4.5)}$$

$$v_3 - v_4 = 50 = 50 \text{ виконана умова (4.4)}$$

Отже, потік не оптимальний. Для поліпшення потоку вибираємо дугу (4,5). Послідовність визначення потенціалу v_4 : $1 \rightarrow 2 \rightarrow 4$. Послідовність визначення потенціалу v_5 : $1 \rightarrow 2 \rightarrow 3 \rightarrow 5$. Напрямок у циклі перерахунку обираємо проти напрямку дуги (4,5). Отже, маємо такий цикл перерахунку U_P :



Визначаємо число δ :

$$\delta = \min\{5, 3, 5 - 2, 5 - 1\} = 3.$$

Перераховуємо потік: $x'_{42} = 5, x'_{23} = 4, x'_{45} = 2, x'_{53} = 0, x'_{ij} = x_{ij}$ if $(i, j) \notin U_P$.

Помічаємо нові значення потоку на рисунку 4.4.

Перевіримо, чи дійсно потік поліпшився. Попереднє значення потоку складає

$$\sum_{(i,j) \in U} c_{ij} x_{ij} = 2 \cdot 20 + 5 \cdot 30 + 1 \cdot 40 + 2 \cdot 10 + 1 \cdot 50 + 5 \cdot 50 + 3 \cdot 10 = 580.$$

Діюче значення потоку складає

$$\sum_{(i,j) \in U} c_{ij} x'_{ij} = 2 \cdot 20 + 5 \cdot 30 + 4 \cdot 40 + 5 \cdot 10 + 1 \cdot 50 + 2 \cdot 50 + 0 \cdot 10 = 550.$$

Значення потоку зменшилось, значить перерахунок виконаний вірно.

Перевіримо оптимальність діючого потоку. Опора потоку $U^x = \{12, 23, 43, 45\}$ є зв'язаний граф. Визначаємо потенціали:

$$\begin{array}{c}
 v_1=0 \\
 \Downarrow \\
 v_2=20 \\
 \Downarrow \\
 v_3=60 \\
 \Downarrow \\
 v_4=10 \\
 \Downarrow \\
 v_5=60
 \end{array}$$

Перевіряємо оптимальність:

$$v_3 - v_1 = 60 > 30 \text{ виконана умова (4.5)}$$

$$v_2 - v_4 = 10 = 10 \text{ виконана умова (4.5)}$$

$$v_3 - v_5 = 0 < 10 \text{ виконана умова (4.3)}$$

Критерій оптимальності виконаний. Отже, потік із значеннями $x_{12}=2$, $x_{23}=4$, $x_{13}=5$, $x_{42}=5$, $x_{43}=1$, $x_{45}=2$, $x_{53}=0$ є оптимальним для заданої сітки. Досягнуте значення мінімуму складає 550.

Зауваження. Значення потоку, при яких досягається мінімум, можуть бути не єдиними. Проте значення мінімуму – єдине.

Розглянемо ще один приклад розв'язання задачі про оптимальний потік, в якому зустрічається вироджений випадок.

Приклад 2. П'ять вершин $I=\{1,2,3,4,5\}$ з'єднані дугами $U=\{12,14,24,25,32,34,45\}$. Задані значення інтенсивностей вершин $d_1=3$, $d_2=0$, $d_3=2$, $d_4=-1$, $d_5=-4$. Пропускна спроможність всіх дуг однакова і складає $r_{ij}=3$. Вартості використання дуг задані наступними значеннями: $c_{12}=10$, $c_{14}=20$, $c_{25}=20$, $c_{24}=5$, $c_{32}=15$, $c_{34}=35$, $c_{45}=10$. Потрібно знайти значення потоку x , який мінімізує значення величини $\sum_{(i,j) \in U} c_{ij} \cdot x_{ij}$.

За умовами задачі складемо сітку (рис. 4.5).

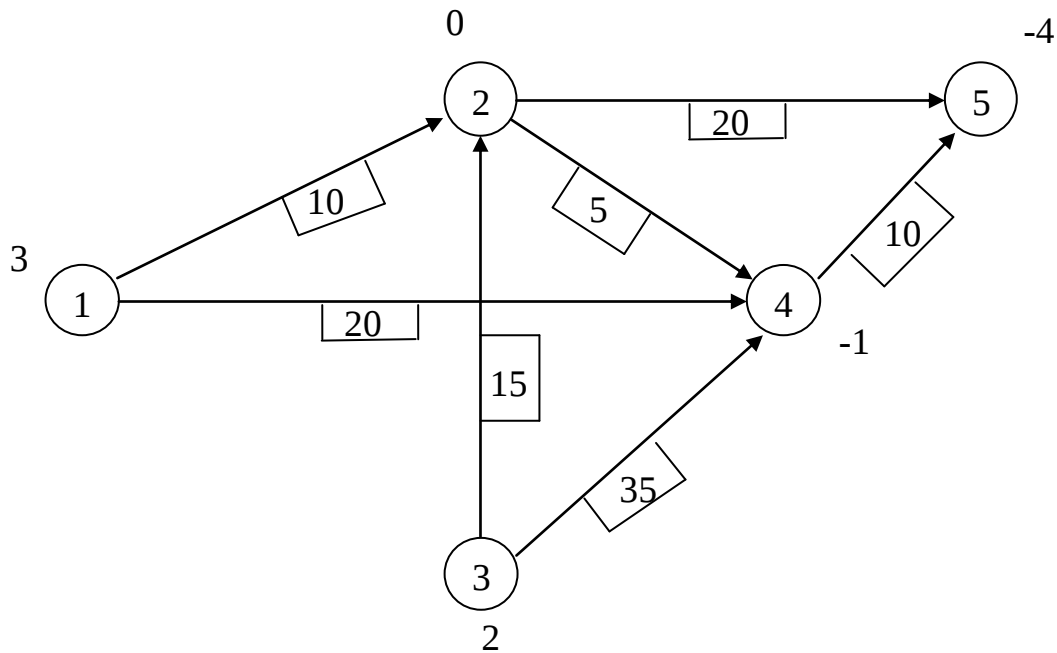


Рисунок 4.5. Сітка до прикладу 2

Побудуємо початковий потік так, щоб виконувались умови (4.1), (4.2) (рис.4.6):

$$x_{12}=2, x_{14}=1, x_{32}=1, x_{34}=1, x_{24}=2, x_{25}=1, x_{45}=3.$$

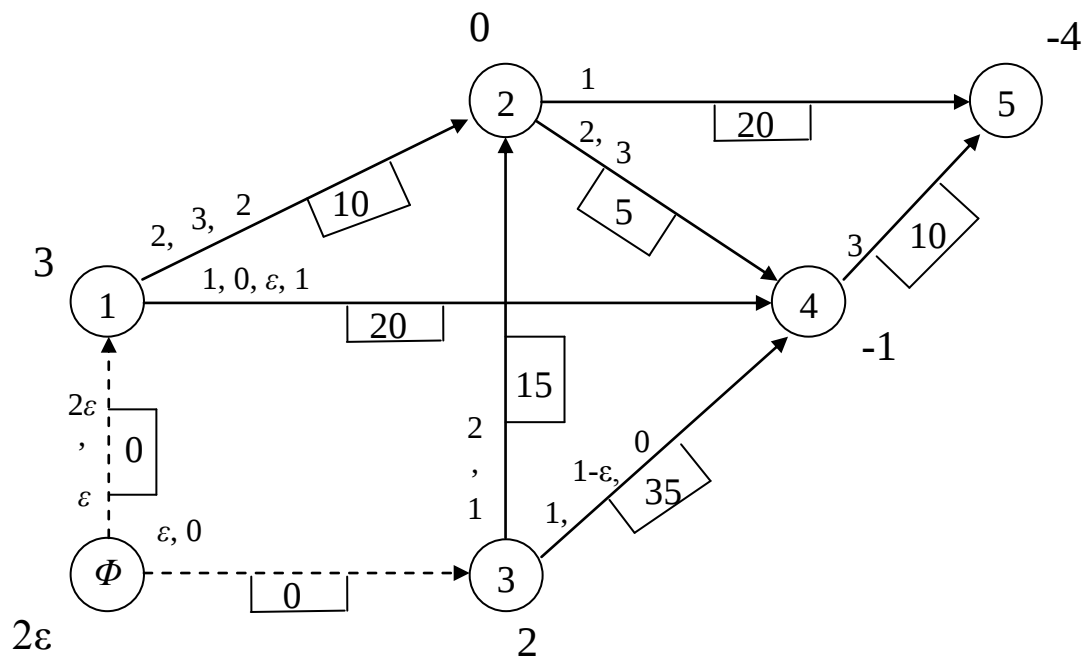


Рисунок 4.6. Розв'язання задачі про оптимальний потік у виродженому випадку

Опора цього потоку $U^x = \{12, 14, 32, 34, 23, 25\}$ є зв'язаний граф. Визначаємо потенціали вершин:

$$\begin{aligned} v_1 &= 0 \\ \Downarrow \\ v_2 &= 10, v_4 = 20 \\ \Downarrow \\ v_3 &= -5, v_5 = 30 \end{aligned}$$

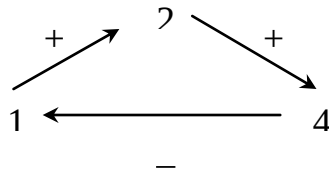
Перевіряємо оптимальність:

$$v_5 - v_4 = 10 = 10 \text{ виконана умова (4.5)}$$

$$v_4 - v_2 = 10 \neq 5 \text{ не виконана умова (4.4)}$$

$$v_4 - v_3 = 25 \neq 35 \text{ не виконана умова (4.4)}$$

Для поліпшення потоку вибираємо дугу (2,4). Побудуємо цикл перерахунку U_P , в якому напрямком обираємо за напрямком дуги (2,4):



Визначаємо число δ :

$$\delta = \min\{3 - 2, 1, 3 - 2\} = 1.$$

Перераховуємо потік: $x'_{24} = 3, x'_{14} = 0, x'_{12} = 3, x'_{ij} = x_{ij}$ if $(i, j) \notin U_P$. Помічаємо нові значення потоку на рисунку 4.6.

Перевіримо, чи дійсно потік поліпшився. Попереднє значення потоку складає

$$\sum_{(i,j) \in U} c_{ij} x_{ij} = 2 \cdot 10 + 1 \cdot 20 + 1 \cdot 20 + 2 \cdot 5 + 1 \cdot 15 + 1 \cdot 35 + 3 \cdot 10 = 150.$$

Діюче значення потоку складає

$$\sum_{(i,j) \in U} c_{ij} x'_{ij} = 3 \cdot 10 + 0 \cdot 20 + 1 \cdot 20 + 3 \cdot 5 + 1 \cdot 15 + 1 \cdot 35 + 3 \cdot 10 = 145.$$

Значить перерахунок виконаний вірно.

Опора потоку $U^x = \{32, 34, 25\}$ є не зв'язаний граф. Отже, визначити потенціали всіх вершин не зможемо. Введемо фіктивну вершину Φ (див. рис. 4.6). З'єднаємо її фіктивними дугами з вершинами, що належать двом різним компонентам зв'язності – вершинами 1 і 3. Зменшимо інтенсивності вершин 1 і 3 на малу величину ε , а інтенсивність фіктивної вершини присвоїмо рівній 2ε . Фіктивним дугам присвоїмо значення ε , нульову вартість і необмежену пропускну спроможність. Опора такого потоку $\tilde{U}^x = \{\Phi 1, \Phi 3, 32, 34, 25\}$ є зв'язаний граф. Визначаємо потенціали:

$$\begin{aligned} v_1 &= 0 \\ \Downarrow \\ v_\Phi &= 0 \\ \Downarrow \\ v_3 &= 0 \\ \Downarrow \\ v_2 &= 15, v_4 = 35 \\ \Downarrow \\ v_5 &= 35 \end{aligned}$$

Перевіряємо оптимальність:

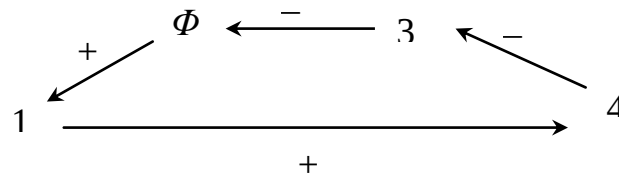
$$v_4 - v_1 = 35 > 20 \text{ не виконана умова (4.3),}$$

$$v_2 - v_1 = 15 > 10 \text{ виконана умова (4.5),}$$

$$v_4 - v_2 = 20 > 5 \text{ виконана умова (4.5),}$$

$v_5 - v_4 = 0 < 10$ не виконана умова (4.5).

Критерій оптимальності не виконаний. Вибираємо для поліпшення потоку дугу (1,4). Побудуємо цикл перерахунку U_P , в якому напрямком обираємо за напрямком дуги (1,4):



Визначаємо число δ :

$$\delta = \min\{3-1, 1, \varepsilon, \infty\} = \varepsilon.$$

Перераховуємо потік: $x'_{14} = \varepsilon$, $x'_{34} = 1 - \varepsilon$, $x'_{\Phi 3} = 0$, $x'_{\Phi 1} = 2\varepsilon$, $x'_{ij} = x_{ij}$ if $(i, j) \notin U_P$.

Помічаємо нові значення потоку на рисунку 4.6.

Перевіримо, чи дійсно потік поліпшився. Попереднє значення потоку складає

$$\sum_{(i,j) \in U} c_{ij} x_{ij} = 3 \cdot 10 + 0 \cdot 20 + 1 \cdot 20 + 3 \cdot 5 + 1 \cdot 15 + 1 \cdot 35 + 3 \cdot 10 = 145.$$

Діюче значення потоку складає

$$\sum_{(i,j) \in U} c_{ij} x'_{ij} = 3 \cdot 10 + \varepsilon \cdot 20 + 1 \cdot 20 + 3 \cdot 5 + 1 \cdot 15 + (1 - \varepsilon) \cdot 35 + 3 \cdot 10 = 145 - 15\varepsilon.$$

Значення потоку зменшилось на малу величину 5ε , значить перерахунок виконаний вірно.

Опора потоку $\tilde{U}^x = \{\Phi 1, 14, 32, 34, 25\}$ є зв'язаний граф. Визначаємо потенціали:

$$\begin{array}{c} v_1 = 0 \\ \Downarrow \\ v_4 = 20 \\ \Downarrow \\ v_3 = -15 \\ \Downarrow \\ v_2 = 0 \\ \Downarrow \\ v_5 = 20 \end{array}$$

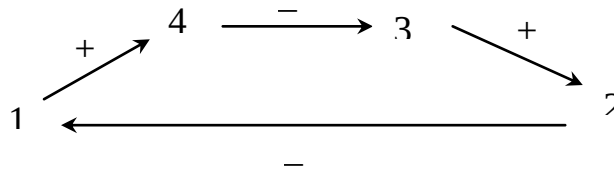
Перевіряємо оптимальність:

$$v_4 - v_2 = 20 > 5 \text{ виконана умова (4.3)}$$

$$v_2 - v_1 = 0 < 10 \text{ не виконана умова (4.5)}$$

$$v_5 - v_4 = 0 < 10 \text{ не виконана умова (4.5)}$$

Критерій оптимальності не виконаний. Вибираємо для поліпшення потоку дугу (1,2). Побудуємо цикл перерахунку U_P , в якому напрямком обираємо проти напрямку дуги (1,2):



Оскільки до циклу перерахунку не належать фіктивні дуги, повернемося до умов задачі прирівняв $\varepsilon=0$.

Визначаємо число δ :

$$\delta = \min\{3, 3-0, 1, 3-2\} = 1.$$

Перераховуємо потік: $x'_{12} = 2, x'_{14} = 1, x'_{43} = 0, x'_{32} = 2, x'_{ij} = x_{ij}$ if $(i, j) \notin U_p$.

Помічаємо нові значення потоку на рисунку 4.6.

Перевіримо, чи дійсно потік поліпшився. Попереднє значення потоку складає

$$\sum_{(i,j) \in U} c_{ij} x_{ij} = 3 \cdot 10 + \varepsilon \cdot 20 + 1 \cdot 20 + 3 \cdot 5 + 1 \cdot 15 + (1 - \varepsilon) \cdot 35 + 3 \cdot 10 = 145 - 15\varepsilon.$$

Діюче значення потоку складає

$$\sum_{(i,j) \in U} c_{ij} x'_{ij} = 2 \cdot 10 + 1 \cdot 20 + 1 \cdot 20 + 3 \cdot 5 + 2 \cdot 15 + 0 \cdot 35 + 3 \cdot 10 = 135.$$

Значення потоку зменшилось, значить перерахунок виконаний вірно.

Опора потоку $\bar{U}^x = \{12, 14, 32, 25\}$ є зв'язаний граф. Визначаємо потенціали:

$$\begin{aligned} v_1 &= 0 \\ \Downarrow \\ v_4 &= 20, v_2 = 10 \\ \Downarrow \\ v_3 &= -5, v_5 = 30 \end{aligned}$$

Перевіряємо оптимальність:

$$v_4 - v_3 = 25 < 35 \text{ виконана умова (4.3),}$$

$$v_5 - v_4 = 10 = 10 \text{ виконана умова (4.5),}$$

$$v_4 - v_2 = 10 > 5 \text{ виконана умова (4.5).}$$

Критерій оптимальності виконаний. Отже, потік із значеннями $x_{12}=2, x_{14}=1, x_{24}=3, x_{25}=1, x_{32}=2, x_{34}=0, x_{45}=3$ є оптимальним для заданої сітки. Досягнуте значення мінімуму складає 135.

Алгоритм розв'язання задачі про оптимальний потік методом потенціалів полягає у наступних кроках:

- Крок 0. Побудувати початковий потік за умовами (4.1), (4.2). Перейти до кроку 1.
- Крок 1. Побудувати опору потоку. Визначити потенціали вершин за формулами (4.6). Якщо критерій оптимальності (4.3), (4.4), (4.5) виконаний, то кінець алгоритму. Інакше перейти до кроку 2.
- Крок 2. Побудувати цикл перерахунку. Виконати перерахунок потоку за формулами (4.8), (4.7). Обчислити значення величини $\sum_{(i,j) \in U} c_{ij} \cdot x_{ij}$. Перейти до кроку 1.

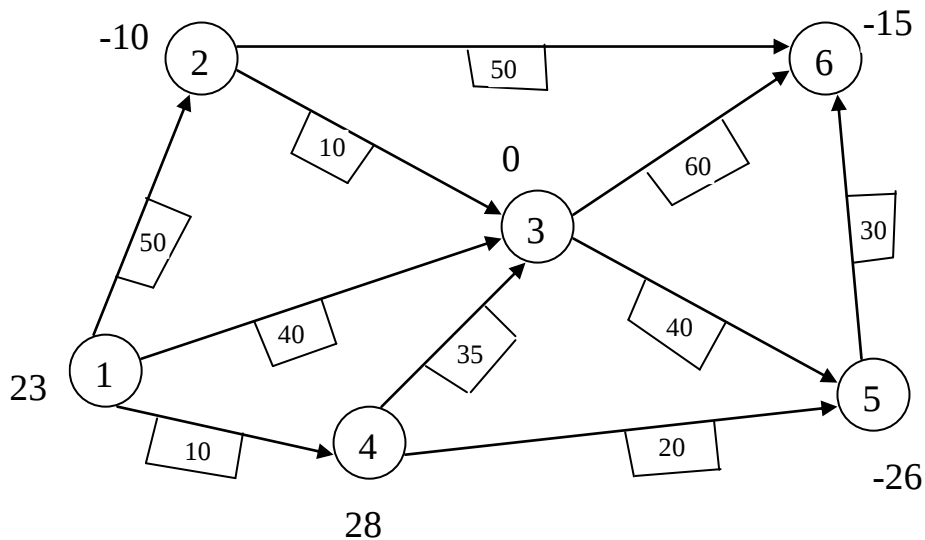
Успішність застосування методу потенціалів залежить від вибору початкового потоку – чим більше початковий потік наближений до оптимального, тим швидше буде знайдений оптимальний потік. На жаль, вироджений випадок, який є складним у реалізації на ЕОМ, зустрічається в розв’язанні задачі про оптимальний потік дуже часто. Обминати його інколи вдається, якщо розпочати розв’язання з іншого початкового потоку.

2. Завдання до роботи

Для даного варіанту сітки скласти та реалізувати на ЕОМ алгоритм відшукування оптимального потоку. Розрахувати значення оптимального потоку за допомогою складеного алгоритму. При складанні алгоритму передбачити можливість зміни вартості будь-якої дуги.

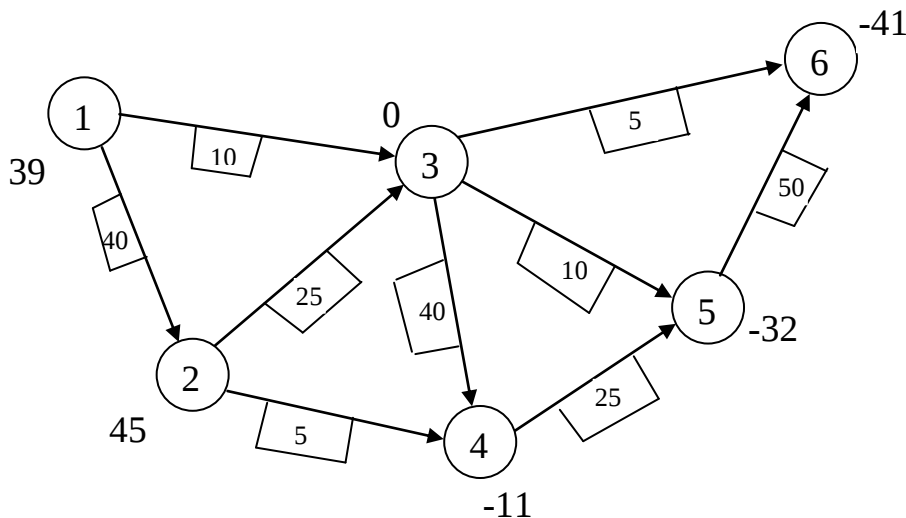
3. Варіанти

1)



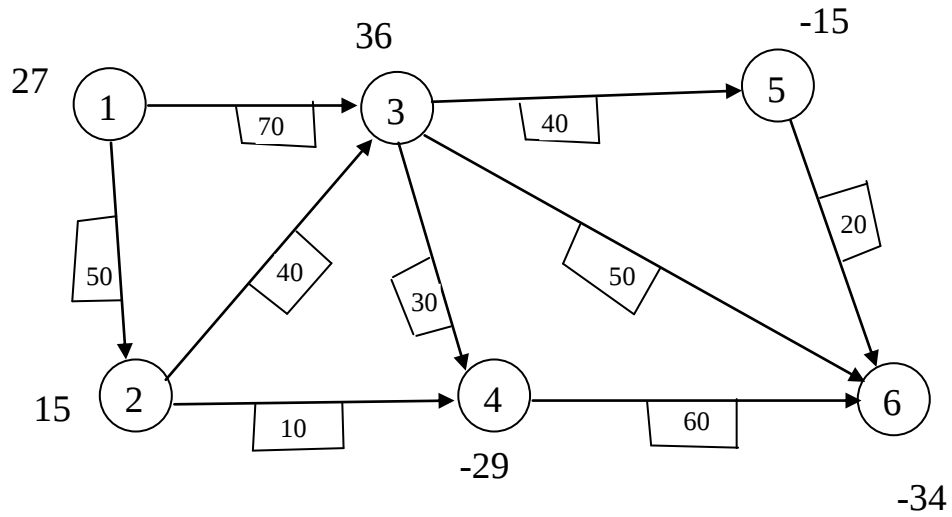
$$r_{12}=20, r_{13}=20, r_{14}=20, r_{26}=30, r_{23}=30, r_{36}=10, r_{35}=10, r_{34}=10, r_{45}=20, r_{56}=20$$

2)



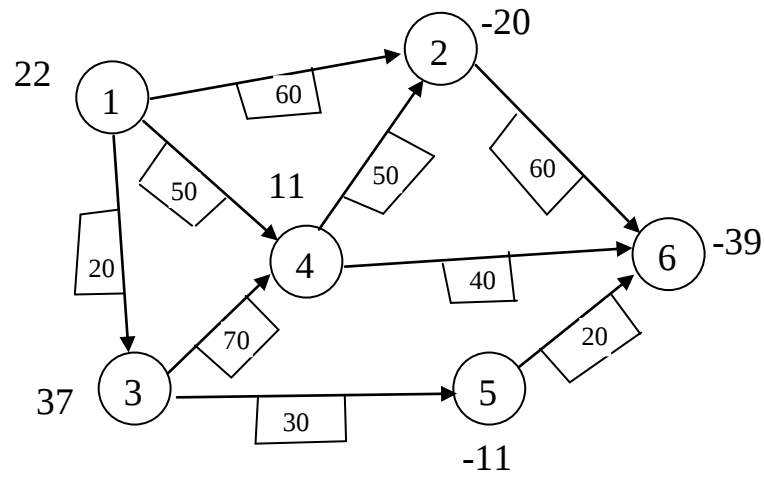
$$r_{13}=40, r_{12}=40, r_{23}=50, r_{36}=30, r_{35}=30, r_{34}=20, r_{24}=50, r_{45}=20, r_{56}=20$$

3)



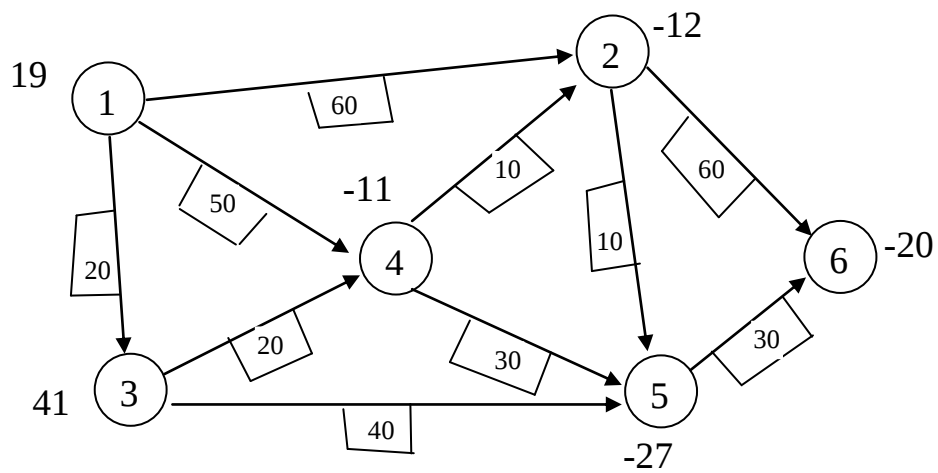
$$r_{13}=40, r_{12}=40, r_{35}=50, r_{34}=50, r_{23}=20, r_{24}=20, r_{46}=30, r_{56}=30, r_{36}=50$$

4)



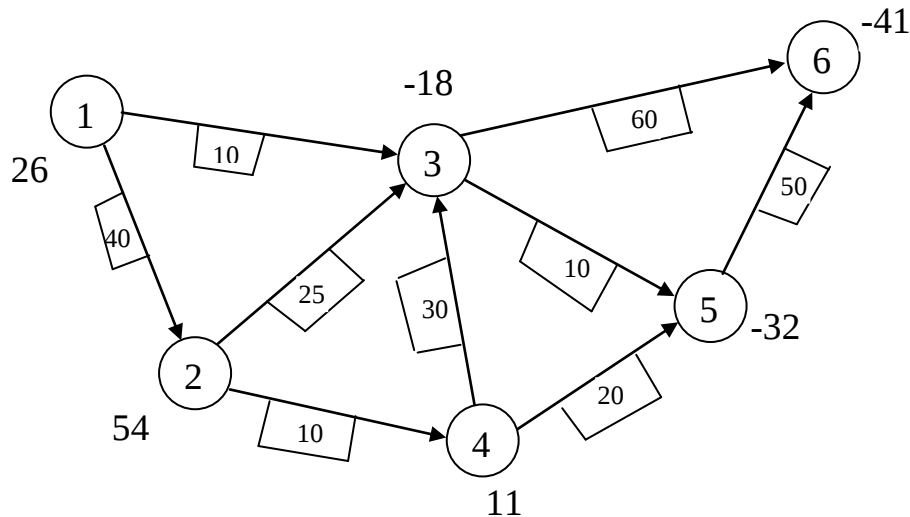
$$r_{12}=30, r_{14}=30, r_{13}=10, r_{26}=20, r_{42}=40, r_{46}=30, r_{45}=30, r_{35}=20, r_{56}=20, r_{34}=40$$

5)



$$r_{12}=30, r_{14}=30, r_{13}=10, r_{26}=20, r_{42}=40, r_{46}=30, r_{45}=30, r_{35}=20, r_{56}=20, r_{34}=40$$

6)



$$r_{13}=40, r_{12}=20, r_{23}=30, r_{24}=10, r_{35}=30, r_{36}=40, r_{43}=50, r_{45}=20, r_{56}=40$$

4. Хід виконання роботи

- 1) уважно розібрати теоретичні відомості щодо задачі про оптимальний потік;
- 2) разом з викладачем вибрати варіант завдання;
- 3) скласти сіткову модель вибраного варіанту завдання;
- 4) скласти алгоритм розв'язання задачі про оптимальний потік використовуючи той засіб програмування, який на Вашу думку найбільш зручний для цієї задачі;
- 5) здійснити перевірку складеного алгоритму на прикладі задачі з наперед відомим результатом розрахунку;
- 6) зробити розрахунок оптимального потоку за даними Вашого варіанту завдання;
- 7) проаналізувати результати і зробити висновки щодо правильності отриманих розрахунків та реалізованого алгоритму;
- 8) оформити звіт.

5. Зміст звіту

- 1) завдання до роботи;
- 2) рисунок сітки до задачі;
- 3) лістинг програми;
- 4) результати розрахунків;
- 5) висновки.

6. Контрольні питання

- 1) У чому полягає задача про оптимальний потік?
- 2) Які умови є необхідними і достатніми для того, щоб потік був оптимальним?
- 3) Який Ви знаєте алгоритм відшукування оптимального потоку?
- 4) Як виконується розрахунок потенціалів у складеному Вами алгоритмі?
- 5) Як будується цикл перерахунку потоку?

- 6) За якими формулами виконується поліпшення неоптимального потоку?
 7) Які труднощі виникли у Вас при реалізації алгоритму на ЕОМ? Як Ви їх подолали?
 8) Чи піддається алгоритм, який Ви склали універсалізації?

7. Література

Ермольев Ю.М., Ляшко И.И., Михалевич В.С., Тюптя В.И., Математические методы исследования операций: Учебн. пособие для вузов. - К.: Вища школа, 1979. – 312с.

ЛАБОРАТОРНА РОБОТА 5 “Сіткове планування робіт”

Мета роботи: навчитися складати та реалізовувати алгоритм сіткового планування робіт.

1. Короткі теоретичні відомості з сіткового планування робіт

Проектом називають сукупність робіт, необхідних для досягнення певної мети. Перед керівником проекту ставиться задача такого планування робіт, при якому час виконання всього проекту - мінімальний. Роботою може бути будь який елемент технологічного процесу, що вимагає затрат часу та матеріальних ресурсів, або тільки затрат часу. Роботи зв'язані між собою, якщо одна з них не може початися, доки інша не є виконаною. Тривалості виконання робіт і зв'язки між роботами являються відомими.

Задача про сіткове планування робіт полягає у тому, щоб знайти мінімально можливий час виконання проекту та скласти сітковий графік виконання робіт.

Сіткова модель проекту складається наступним чином. Подія полягає у тому, що на даний момент часу є виконаними певні роботи. Кожній події відповідає вершина i . Якщо здійсненню події j передуює подія i , то вершини i та j з'єднуються дугою. Дугам відповідають роботи. Дузі (i, j) присвоюється значення часу виконання відповідної роботи t_{ij} . Завжди існує подія “початок виконання проекту”, яка не має попередніх подій, і подія “кінець виконання проекту”, яка не має наступних подій. Поставимо у відповідність події “початок виконання проекту” вершину 1 та у відповідність події “кінець виконання проекту” - вершину n . Будь-який путь із вершини 1 до вершини n відповідає послідовності технологічних операцій, необхідних для виконання проекту. Час виконання проекту не може бути меншим за сумарний час виконання робіт уздовж “найдовшого” путі. Тому мінімально можливий час визначається максимальною “довжиною” путі:

$$Time = \max_{U_P} \sum_{(i,j) \in U_P} t_{ij}, \quad (5.1)$$

де U_P – будь-який путь в сітковій моделі.

Роботи, які належать “найдовшому” путі в сітковій моделі проекту, називають *критичними*. Якщо критична робота виконана за будь-яких причин невчасно, то і весь проект буде виконаний невчасно. Тому найбільш пильно керівник проекту повинен слідкувати за виконанням критичних робіт.

Кожній події i поставимо у відповідність два моменти – ранній $T_{\min}(i)$ та пізній $T_{\max}(i)$ строки його здійснення. Ранній строк виконання події 1 покладемо рівним 0:

$$T_{\min}(1)=0.$$

Ранні строки здійснення усіх інших подій визначаються послідовно від першої вершини до останньої за формулою:

$$T_{\min}(j) = \max_i(T_{\min}(i) + t_{ij}). \quad (5.2)$$

Після того як визначений ранній строк здійснення останньої події пізній строк цієї події приймається рівним ранньому строку:

$$T_{\max}(n) = T_{\min}(n).$$

Потім розраховуються пізні строки здійснення усіх інших подій за формулою:

$$T_{\max}(j) = \min_i(T_{\max}(i) - t_{ji}). \quad (5.3)$$

Робота (i, j) є критичною, якщо виконані наступні умови:

$$T_{\min}(i) = T_{\min}(j), \quad (5.4)$$

$$T_{\max}(i) = T_{\max}(j), \quad (5.5)$$

$$T_{\min}(j) - T_{\min}(i) = T_{\max}(j) - T_{\max}(i) = t_{ij}. \quad (5.6)$$

Критичні роботи виконуються одна за одною без затримок. Некритичні роботи мають деякий резерв часу для їх виконання. *Повний резерв часу* – це максимально можливий запас часу для виконання роботи, який розраховується за формулою:

$$R_{ij}^{\max} = T_{\max}(j) - T_{\min}(i) - t_{ij}. \quad (5.7)$$

Вільний резерв часу – це мінімальний запас часу, який розраховується за умови, що попередня і наступна подія здійснюються у свої ранні строки:

$$R_{ij}^{\min} = T_{\min}(j) - T_{\min}(i) - t_{ij}. \quad (5.8)$$

Розглянемо на прикладі розв’язання задачі про сіткове планування робіт.

Приклад 1. Перелік робіт та тривалості їх виконання подані у таблиці 5.1. Потрібно виконати оптимальне планування робіт.

Таблиця 5.1

Робота	1; 2	1; 3	2; 3	2; 4	2; 5	3; 4	3; 5	4; 5	4; 6	5; 6
Час виконання роботи	1	5	10	15	18	3	11	8	20	10

За даними таблиці 5.1 складемо сіткову модель проекту (рис. 5.1). Для зручності виконання розрахунків поділимо вершину на чотири частини так, як на рисунку 5.2. У верхній частині вершини будемо позначати номер події, у лівій частині – ранній строк здійснення подій, у правій частині – пізній строк

здійснення події, у нижній частині – різницю між пізнім та раннім строками здійснення події.

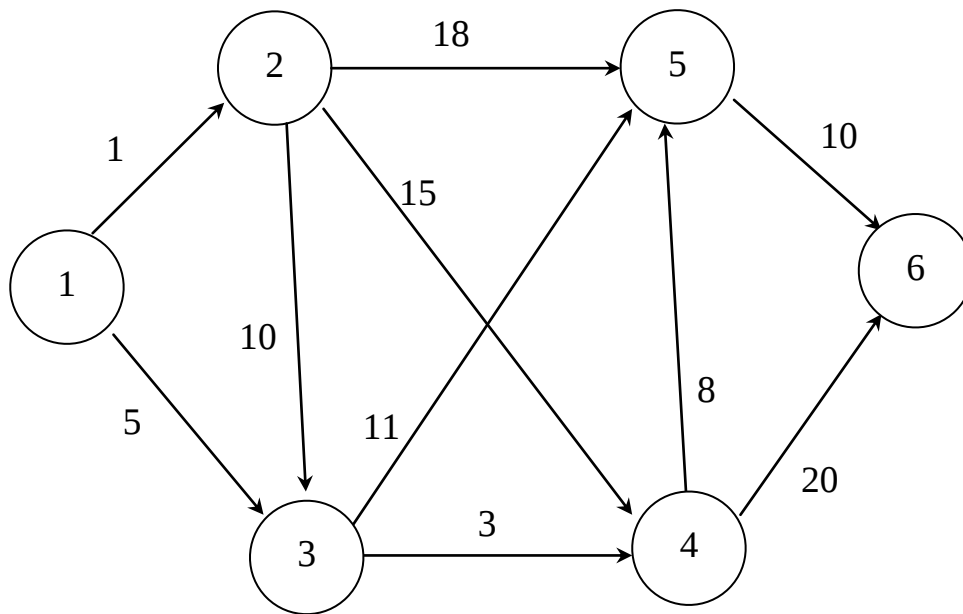


Рисунок 5.1. Сіткова модель до прикладу 1

Починаємо розрахунки з першої вершини – приймаємо ранній строк здійснення першої події рівним нулю (див. рисунок 5.2):

$$T_{min}(1)=0.$$

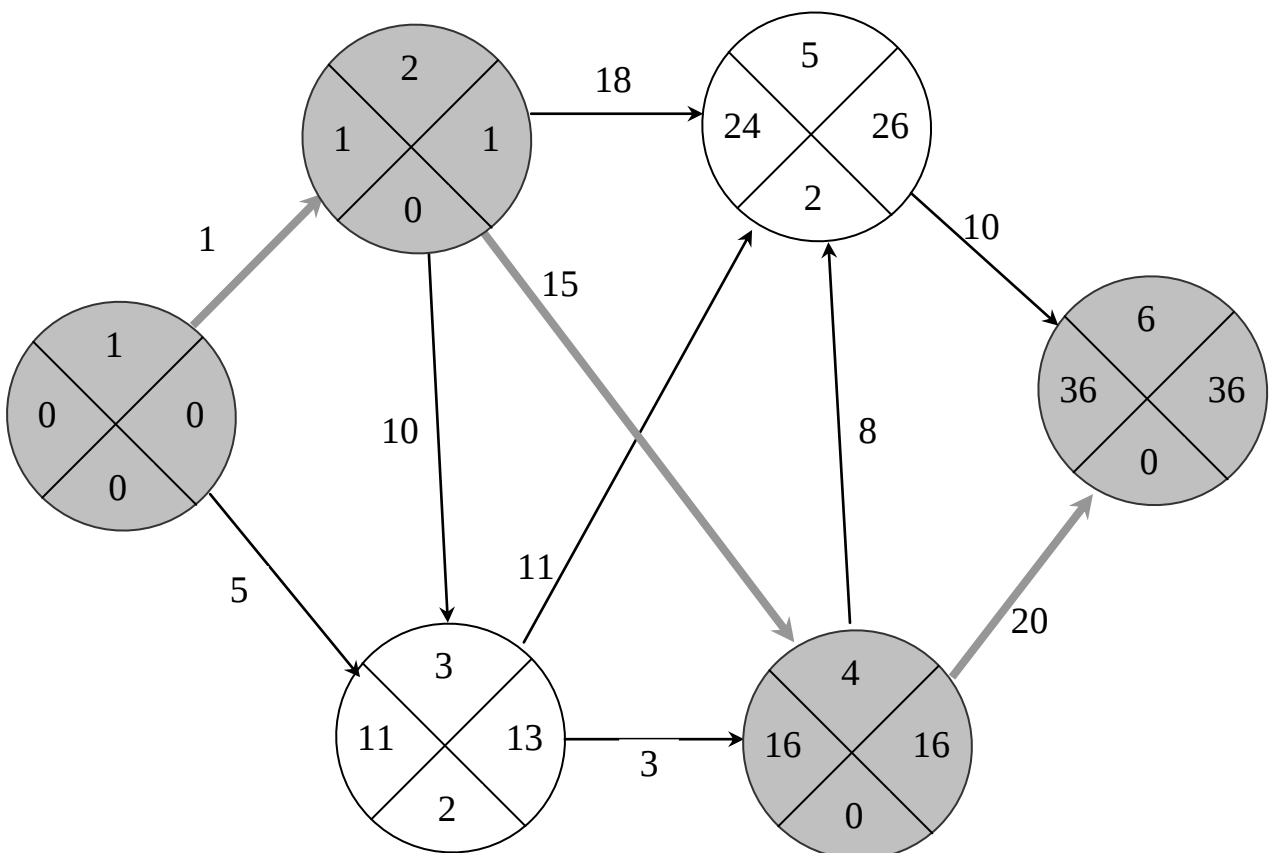


Рисунок 5.2. Розв'язання задачі про сіткове планування робіт

Розраховуємо послідовно ранні строки інших вершин за формулою (5.2):

$$\begin{aligned}T_{\min}(2) &= \max(T_{\min}(1) + t_{12}) = 1, \\T_{\min}(3) &= \max(T_{\min}(1) + t_{13}, T_{\min}(2) + t_{23}) = 11, \\T_{\min}(4) &= \max(T_{\min}(2) + t_{24}, T_{\min}(3) + t_{34}) = 16, \\T_{\min}(5) &= \max(T_{\min}(2) + t_{25}, T_{\min}(3) + t_{35}, T_{\min}(4) + t_{45}) = 24, \\T_{\min}(6) &= \max(T_{\min}(4) + t_{46}, T_{\min}(5) + t_{56}) = 36.\end{aligned}$$

Приймаємо, що пізній строк здійснення останньої події рівний ранньому строку цієї події:

$$T_{\max}(6) = T_{\min}(6).$$

Розраховуємо послідовно пізні строки інших вершин за формулою (5.3):

$$\begin{aligned}T_{\max}(5) &= \min(T_{\max}(6) - t_{56}) = 26, \\T_{\max}(4) &= \min(T_{\max}(6) - t_{46}, T_{\max}(5) - t_{56}) = 16, \\T_{\max}(3) &= \min(T_{\max}(5) - t_{35}, T_{\max}(4) - t_{34}) = 13, \\T_{\max}(2) &= \min(T_{\max}(5) - t_{25}, T_{\max}(3) - t_{23}, T_{\max}(4) - t_{24}) = 1, \\T_{\max}(1) &= \min(T_{\max}(2) - t_{12}, T_{\max}(3) - t_{13}) = 0.\end{aligned}$$

Розраховуємо різницю пізнього та раннього строків здійснення кожної події і заносимо це значення у нижню частину вершини (див. рисунок 5.2). Якщо різниця пізнього та раннього строків здійснення події рівна нулю (умови (5.4), (5.5)), то подія є критичною – відмічаємо вершину сірим кольором. Для дуг, що з'єднують відмічені вершини, перевіряємо виконання умови (5.6). Якщо умова (5.6) виконана, то відмічаємо дугу сірим кольором, а відповідну роботу запам'ятовуємо як критичну.

Для некритичних робіт розраховуємо резерв часу повний та вільний за формулами (5.7), (5.8):

$$\begin{aligned}R_{13}^{\max} &= T_{\max}(3) - T_{\min}(1) - t_{13} = 8 & R_{13}^{\min} &= T_{\min}(3) - T_{\min}(1) - t_{13} = 6 \\R_{23}^{\max} &= T_{\max}(3) - T_{\min}(2) - t_{23} = 2 & R_{23}^{\min} &= T_{\min}(3) - T_{\min}(2) - t_{23} = 0 \\R_{34}^{\max} &= T_{\max}(4) - T_{\min}(3) - t_{34} = 2 & R_{34}^{\min} &= T_{\min}(4) - T_{\min}(3) - t_{34} = 2 \\R_{35}^{\max} &= T_{\max}(5) - T_{\min}(3) - t_{35} = 4 & R_{35}^{\min} &= T_{\min}(5) - T_{\min}(3) - t_{35} = 2 \\R_{45}^{\max} &= T_{\max}(5) - T_{\min}(4) - t_{45} = 2 & R_{45}^{\min} &= T_{\min}(5) - T_{\min}(4) - t_{45} = 0 \\R_{56}^{\max} &= T_{\max}(6) - T_{\min}(5) - t_{56} = 2 & R_{56}^{\min} &= T_{\min}(6) - T_{\min}(5) - t_{56} = 2\end{aligned}$$

Перебіг подій у часі зручно представити у вигляді графіка (рис. 5.3). Темним кольором відмічені критичні роботи. Вони завжди у часі слідуєть одна за одною. Події 1, 2, 4 та 6, які являються критичними, також відмічені темним кольором. Час, в який повинні здійснитися ці події, є для керівника проекту жорстко заданим. Затримка у здійсненні критичної події означає, що весь комплекс робіт не буде виконаний у мінімальний строк.

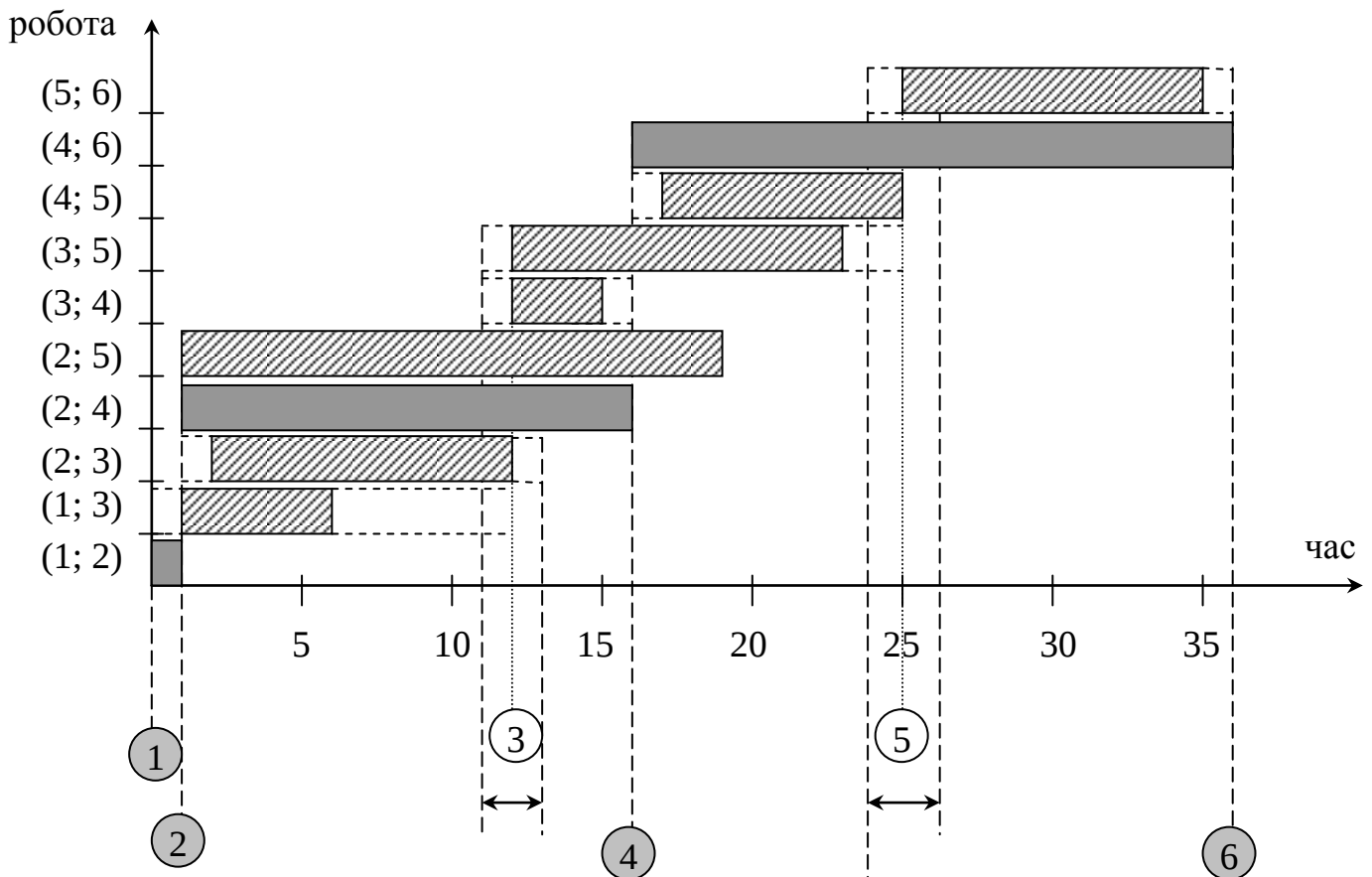


Рисунок 5.3. Графік виконання робіт

Події 3 та 5 являються некритичними, час їх здійснення знаходиться в межах від мінімально можливого до максимально можливого. Так, здійснення події 3 обмежено моментами часу $t=11$ та $t=13$. Подія 3 здійсниться в момент часу $t=11$ за умови, що виконання роботи (2,3) розпочато в найбільш ранній свій строк $t=1$, і здійсниться в моменту часу $t=13$, якщо виконання роботи (2,3) розпочато в найбільш пізній свій строк $t=3$. Але здійснення події 3 не може бути відкладене на строк, більший за момент часу $t=13$, тому що це спричинить запізнення початку роботи (3,4) і значить запізнення здійснення критичної події 4. Здійснення події 5 обмежено моментами часу $t=24$ та $t=26$ (див. рис. 5.3).

2. Завдання до роботи

Для даного варіанту скласти список робіт, задати тривалість виконання кожної операції та побудувати сіткову модель. Реалізувати на ЕОМ алгоритм сіткового планування робіт. Розрахувати мінімальний час виконання проекту та визначити список критичних робіт. Для некритичних робіт визначити повний та вільний резерв часу. Скласти графік виконання робіт. При складанні алгоритму передбачити можливість зміни тривалості будь-якої роботи.

3. Варіанти

- 1) Планування навчального процесу;
- 2) Планування будівництва житлового будинку;

- 3) Планування збирання ЕОМ;
- 4) Планування збирання автомобіля (літака, корабля та ін.);
- 5) Планування виконання дипломного проекту;
- 6) Планування створення інформаційної системи.

4. Хід виконання роботи

- 1) уважно розібрати теоретичні відомості щодо сіткового планування робіт;
- 2) разом з викладачем вибрати варіант завдання;
- 3) скласти алгоритм розв'язання задачі про сіткове планування робіт використовуючи той засіб програмування, який на Вашу думку найбільш зручний для цієї задачі;
- 4) знайти мінімальний час виконання проекту та список критичних робіт;
- 5) розрахувати повний та вільний резерви часу для некритичних робіт;
- 6) скласти графік виконання робіт;
- 7) проаналізувати результати і зробити висновки щодо правильності отриманих розрахунків та реалізованого алгоритму;
- 8) оформити звіт.

5. Зміст звіту

- 1) завдання до роботи;
- 2) рисунок сітки до задачі;
- 3) лістинг програми;
- 4) результати розрахунків;
- 5) графік планування робіт;
- 6) висновки.

6. Контрольні питання

- 1) У чому полягає задача про сіткове планування робіт?
- 2) Які алгоритми Ви знаєте розв'язання задачі про сіткове планування робіт?
- 3) Поясніть, як виконується сіткове планування робіт?
- 4) Як розрахувати пізній та ранній строки виконання робіт? Який зміст вони мають?
- 5) Як розрахувати повний та вільний резерви виконання робіт? Який зміст вони мають?
- 6) Як складається графік виконання робіт? Яку інформацію він представляє?
- 7) Які труднощі виникли у Вас при реалізації алгоритму на ЕОМ? Як Ви їх подолали?
- 8) Чи піддається алгоритм, який Ви склали, універсалізації?

7. Література

Деордица Ю.С., Нефедов Ю.М. Исследование операций в планировании и управлении. – К., Вища школа. – 1991. – 270с.

ЛАБОРАТОРНА РОБОТА 6

“Дослідження властивостей сітки Петрі матричним способом”

Мета роботи: навчитися реалізовувати сітки Петрі та досліджувати їх властивості.

1. Короткі теоретичні відомості з теорії сіток Петрі

Сітка Петрі є засобом формального опису процесів функціонування дискретних систем. В дискретній системі зміни її стану трапляються в особливі моменти часу, коли виникають умови для здійснення події. Здійснення події означає зміну стану системи, а значить, виникнення або не виникнення умов для інших подій. Процес функціонування дискретної системи – це упорядкована в часі послідовність подій.

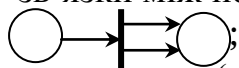
Приклад 1. Розглянемо, наприклад, процес обслуговування клієнтів звичайним банкоматом. Подія перша полягає у тому, що клієнт підходить до банкомату. Подія друга - клієнт розпочав виконувати потрібні йому операції. Подія третя - клієнт закінчив виконувати потрібні йому операції. Подія четверта – клієнт залишив банкомат. Якщо у момент, коли клієнт з'явився перед банкоматом, банкомат зайнятий іншим клієнтом, то клієнт очікує звільнення банкомату. Умови, що передують виконанню певної події називають *передумовами*. Умови, які є результатом здійснення події, називають *післяумовами*. Так, передумовами події „клієнт розпочав виконувати операції” являються умови а) „банкомат вільний” та б) „у черзі клієнтів є клієнт”. Післяумовами події „клієнт розпочав виконувати операції” являється умова „клієнт обслуговується”. Передумовами події „клієнт закінчив виконувати операції” являється умова „клієнт обслуговується”. Післяумовами цієї події являються умови а) „банкомат вільний” та б) „є обслугований клієнт”. Складемо таблицю взаємозв'язків між подіями та умовами (табл. 6.1):

Таблиця 6.1

Таблиця взаємозв'язків між подіями та умовами

Подія	Передумови події	Післяумови події
„клієнт підійшов до банкомату”	—	„у черзі клієнтів є клієнт”
„клієнт розпочав виконувати операції”	„у черзі клієнтів є клієнт”; „банкомат вільний”	„клієнт обслуговується”
„клієнт закінчив виконувати операції”	„клієнт обслуговується”	„банкомат вільний”; „є обслугований клієнт”
„клієнт залишив банкомат”	„є обслугований клієнт”	—

При графічному представленні сітки Петрі користуються такими позначеннями:

- події зображуються планками, які називають *переходами*, - ;
- умови для виникнення подій зображуються кругами, які називають *позиціями*, - ;
- зв'язки між позиціями та переходами зображуються дугами - ;
- виконання (або не виконання) умови зображується наявністю (або відсутністю) *фішки* в позиції -  .

Отже, процес обслуговування клієнтів банкоматом представляється сіткою Петрі, яка зображена на рисунку 6.1.

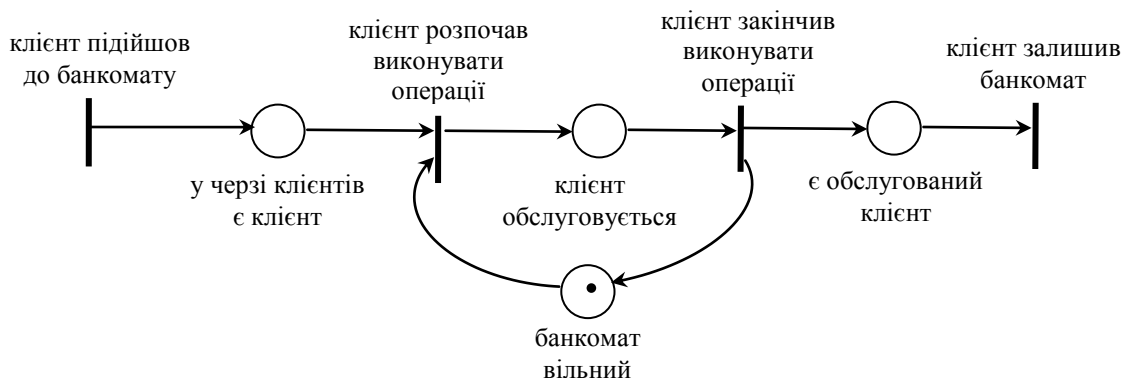


Рисунок 6.1 Сітка Петрі, яка представляє процес обслуговування клієнтів банкоматом

Коли спрацює перехід „клієнт підійшов до банкомату”, в позиції „у черзі клієнтів є клієнт” з’явиться фішка. Наявність фішок у позиціях „у черзі клієнтів є клієнт” та „банкомат вільний” означає, що виконані умови для події „клієнт розпочав виконувати операції”. При запуску переходу „клієнт розпочав виконувати операції” фішки з позицій „у черзі клієнтів є клієнт” та „банкомат вільний” видаляються, а в позиції „клієнт обслуговується” фішка з’являється. Наявність фішки у позиції „клієнт обслуговується” означає, що виконані умови для події „клієнт закінчив виконувати операції”. При запуску цього переходу фішка з позиції „клієнт обслуговується” видаляється, а в позиції „банкомат вільний” та „є обслугований клієнт” фішки додаються. Наявність фішки в позиції „є обслугований клієнт” являється умовою запуску переходу „клієнт залишає банкомат”. При запуску цього переходу фішка з позиції „є обслугований клієнт” видаляється.

На перший погляд здається, що умова „банкомат вільний” являється зайвою. Але наявність умови „банкомат вільний” забезпечує, що обслуговування клієнтів здійснюється по-одному.

Випадковий характер процесів, які моделюються, проявляється у випадковому запуску переходів, для яких виконані передумови:

Якщо виконані передумови зразу для декількох переходів, то послідовність їх запуску визначається випадком.

В процесі обслуговування клієнтів банкоматом можемо спостерігати, наприклад, таку послідовність переходів:

клієнт підійшов до банкомату;
 клієнт розпочав виконувати операції;
 клієнт закінчив виконувати операції;
 клієнт залишив банкомат;
 клієнт підійшов до банкомату.

Наступна послідовність переходів теж являється можливою:

клієнт підійшов до банкомату;
 клієнт розпочав виконувати операції;
 клієнт підійшов до банкомату;
 клієнт закінчив виконувати операції;
 клієнт підійшов до банкомату;
 клієнт залишив банкомат;
 клієнт розпочав виконувати операції;
 клієнт закінчив виконувати операції;
 клієнт залишив банкомат;
 клієнт розпочав виконувати операції;
 клієнт закінчив виконувати операції;
 клієнт залишив банкомат.

У першому випадку клієнтам не доводиться очікувати звільнення банкомату, а у другому спостерігається черга клієнтів до двох чоловік.

Для математичного описування процесів, що відбуваються в сітці Петрі необхідно перейти від текстових назв переходів та позицій до нумерованих їх назв (рис. 6.2).

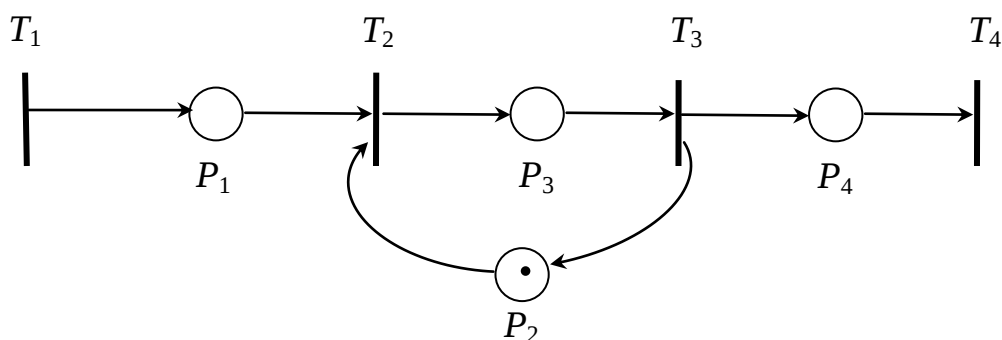


Рисунок 6.2 Формалізована сітка Петрі, яка представляє процес обслуговування клієнтів банкоматом

Зв'язки між позиціями та переходами зручно представити за допомогою матриці входів та матриці виходів. У відповідність кожному j -тому рядочку матриці поставимо перехід T_j , а кожному i -тому стовпчику – позицію P_i . Матрицю входів D^- складемо з елементів D_{ji}^- , які дорівнюють кількості зв'язків між позицією P_i та переходом T_j . А матрицю виходів D^+ – з елементів D_{ji}^+ , які дорівнюють кількості зв'язків між переходом T_j та позицією P_i .

Складемо, наприклад, матриці входів та виходів для сітки Петрі, представленої на рисунку 6.2:

$$D^- = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad D^+ = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

Кількість фішок в позиціях складає вектор, який називається *маркіруванням*. В графічному представленні сітки Петрі зображується тільки початкове маркірування. В процесі функціонування сітки Петрі маркірування змінюється за простими правилами:

Якщо у вхідних позиціях переходу T_j є фішки у кількості рівній кількості зв'язків, то виконана *умова запуску переходу T_j* .

Якщо умова запуску переходу T_j виконана, то виконується *запуск переходу*: з усіх вхідних позицій переходу T_j фішки видаляються у кількості рівній кількості зв'язків, а в усі вихідні позиції переходу T_j фішки додаються у кількості рівній кількості зв'язків

Так, для нашого прикладу початкове маркірування представляється вектором $M=(0,1,0,0)$. В результаті послідовного запуску переходів $T_1 - T_2 - T_1 - T_3 - T_1 - T_4 - T_2 - T_3 - T_4$ маркірування сітки Петрі змінюється наступним чином:

$$\begin{aligned} & (0 \ 1 \ 0 \ 0) \xrightarrow{T_1} (1 \ 1 \ 0 \ 0) \xrightarrow{T_2} (0 \ 0 \ 1 \ 0) \xrightarrow{T_1} (1 \ 0 \ 1 \ 0) \xrightarrow{T_3} \\ & \xrightarrow{T_3} (1 \ 1 \ 0 \ 1) \xrightarrow{T_1} (2 \ 1 \ 0 \ 1) \xrightarrow{T_4} (2 \ 1 \ 0 \ 0) \xrightarrow{T_2} (1 \ 0 \ 1 \ 0) \xrightarrow{T_3} \\ & \xrightarrow{T_3} (1 \ 1 \ 0 \ 1) \xrightarrow{T_4} (1 \ 1 \ 0 \ 0) \xrightarrow{T_2} (0 \ 0 \ 1 \ 0) \xrightarrow{T_3} (0 \ 1 \ 0 \ 1) \xrightarrow{T_4} (0 \ 1 \ 0 \ 0) \end{aligned}$$

Введення матриці входів та матриці виходів дозволяє сформулювати правила функціонування сітки Петрі у матричному вигляді:

Якщо кожна компонента вектора маркірування M_i не менша за елемент матриці входів D^-_{ji} , то умова запуску переходу T_j виконана.

Якщо умова запуску переходу T_j виконана, то виконується запуск переходу: з кожної компоненти вектора маркірування M_i віднімається значення матриці входів D^-_{ji} і додається значення матриці виходів D^+_{ji} .

Іншими словами, якщо вектор маркірування не менший за j -рядочок матриці входів, то умова запуску переходу T_j виконана. якщо умова запуску переходу T_j виконана, то з вектора маркірування віднімається j -рядочок матриці входів і додається j -рядочок матриці виходів.

Визначення j -рядочка матриці математично виражається добутком вектора e_j , всі компоненти якого рівні нулю, а j -компонента рівна 1, і матриці. Наприклад, для переходу T_3 (рис. 6.2) маємо:

$$e_3 D^- = (0 \ 1 \ 0 \ 0) \cdot \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = (1 \ 1 \ 0 \ 0).$$

Отже, у векторному вигляді запуск переходу здійснюється за умови:

$$M \geq e_j \cdot D^- . \quad (6.1)$$

А результат запуску переходу знаходиться за формулою:

$$M' = M - e_j D^- + e_j D^+ = M + e_j (D^+ - D^-) = M + e_j D . \quad (6.2)$$

Матрицю $D = D^+ - D^-$ називають *матрицею змінювань*.

Припустимо, що потрібно знайти результат запуску послідовності переходів $T_1 - T_2 - T_1$:

$$M_1 = M + e_1 D ,$$

$$M_2 = M_1 + e_2 D = M + e_1 D + e_2 D = M + (e_1 + e_2) D ,$$

$$M_3 = M_2 + e_1 D = M + (e_1 + e_2) D + e_1 D = M + (2e_1 + e_2) D = M + vD$$

Зауважимо, що вектор запусків переходів $v = 2e_1 + e_2$ складається з компонент (2,1,0, 0). Якщо згадати послідовність переходів, яка запускала ($T_1 - T_2 - T_1$), то стане ясно, що *вектор запусків переходів* складається з компонентів, що дорівнюють кількості запусків відповідних переходів:

$$v_i = \sum_j k_j e_j , \quad (6.3)$$

де k_j – кількість переходів T_j у заданій послідовності переходів.

Наприклад, результат запуску послідовності переходів $T_1 - T_2 - T_1 - T_3 - T_1 - T_4 - T_2 - T_3 - T_4 - T_2 - T_3 - T_4$, що розглядалася вище, може знайдений матричним способом так:

$$v = (3 \ 3 \ 3 \ 3) ,$$

$$D = D^+ - D^- = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} - \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -1 & -1 & 1 & 0 \\ 0 & 1 & -1 & 1 \\ 0 & 0 & 0 & -1 \end{pmatrix} ,$$

$$M' = M + vD = (0 \ 1 \ 0 \ 0) + (3 \ 3 \ 3 \ 3) \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ -1 & -1 & 1 & 0 \\ 0 & 1 & -1 & 1 \\ 0 & 0 & 0 & -1 \end{pmatrix} =$$

$$= (0 \ 1 \ 0 \ 0) + (0 \ 0 \ 0 \ 0) = (0 \ 1 \ 0 \ 0)$$

Приклад 2. Розглянемо приклад моделювання сіткою Петрі процесу виникнення та усунення несправностей. Деякий виробничий процес обслуговують три пристрої, які час від часу страждають від поломки. Якщо є запасний справний пристрій, то поламаний пристрій вилучається з виробничого процесу і відправляється на ремонтування, а замість нього у виробничий процес встановлюється запасний пристрій. Після ремонту пристрій знову може бути використаний у виробничому процесі.

Виділимо множину подій:

- | виникла поломка в одному з пристроїв;
- | заміна несправного пристрою на відремонтований;
- | ремонт пристрою.

Подія „виникла поломка в одному з пристроїв” здійснюється за умови, що є працюючий пристрій. Подія „заміна несправного пристрою на відремонтований” здійснюється за умови, що є несправний пристрій і є відремонтований пристрій. Отже, множина умов:

- є працюючий пристрій;
- є зламаний пристрій;
- є відремонтований пристрій;
- є вилучений з виробництва пристрій;

З’єднуємо події та умови і отримуємо сітку Петрі, яка представлена на рисунку 6.3.

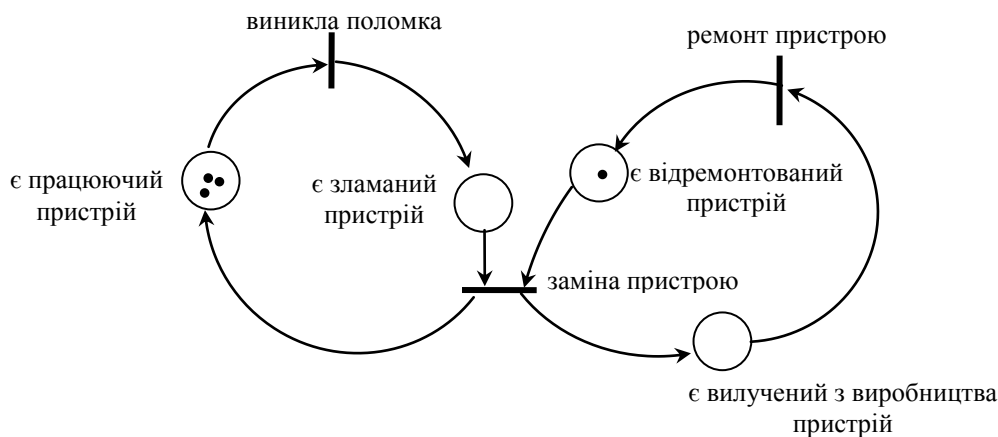


Рисунок 6.3 Сітка Петрі, яка представляє процес виникнення та усунення несправностей

Аналітичне дослідження сіток Петрі дозволяє з'ясувати тільки загальні властивості систем, які вони описують. Такими загальними властивостями являються *k-обмеженість*, *досяжність*, *зберігання* та *живучість*.

Якщо кількість фішок в будь-якій позиції сітки Петрі не перевищує k штук, то сітка являється ***k - обмеженою***. Наприклад, сітка Петрі, яка представлена на рисунку 6.4, являється 4-обмеженою.

В сітках Петрі є маркірування, що не можуть з'явитися ні при якій послідовності подій. Такі маркірування являються недосяжними (принаймні з даного початкового маркірування), всі інші маркірування - досяжні. Наприклад, в сітці Петрі, представлений на рисунку 6.1, в позиції „банкомат вільний” неможлива поява більше одної фішки. Тобто, якщо на початку моделювання існував один банкомат, то в процесі моделювання банкоматів не може стати два, три чи більше.

Досяжністю сітки Петрі називається множина досяжних маркірувань. Нехай потрібно з'ясувати, чи являється маркірування M' досяжним. Складемо матричне рівняння

$$M' = M + v \cdot D, \quad (6.4)$$

де M – відоме початкове маркірування, D – матриця змінювань, v – невідомий вектор запусків переходів. Якщо знайдений з матричного рівняння вектор v має від’ємні або нецілі компоненти, то ясно, що маркірування M' не може бути досяжним. Якщо усі компоненти знайденого вектора v невід’ємні цілі, то потрібне додаткове дослідження досяжності маркірування M' , оскільки воно може бути в цьому випадку як досяжним, так і недосяжним. Існує класичний приклад для доведення того, що існування невід’ємного цілого вектора запуску переходів, що задовольняє рівнянню (6.4), являється тільки необхідною, але не достатньою умовою.

Приклад 3. Спробуємо з’ясувати матричним способом, чи являється маркірування $M'=(0,0,0,1)$ досяжним для сітки Петрі, яка представлена на рисунку 6.4.

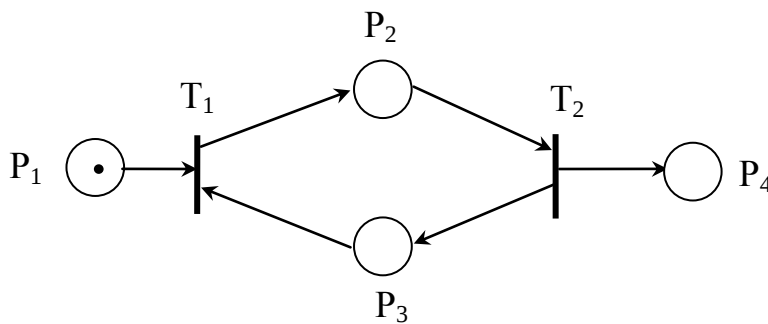


Рисунок 6.4 Сітка Петрі до прикладу 3

Складемо матрицю змінювань

$$D = D^+ - D^- = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} - \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} = \begin{pmatrix} -1 & 1 & -1 & 0 \\ 0 & -1 & 1 & 1 \end{pmatrix}$$

Розв’яжемо матричне рівняння (6.4):

$$M' = M + v \cdot D \Rightarrow (0 \ 0 \ 0 \ 1) = (1 \ 0 \ 0 \ 0) + (v_1 \ v_2) \cdot \begin{pmatrix} -1 & 1 & -1 & 0 \\ 0 & -1 & 1 & 1 \end{pmatrix} \Rightarrow$$

$$\Rightarrow (-1 \ 0 \ 0 \ 1) = (-v_1 \ v_1 - v_2 \ -v_1 + v_2 \ v_2) \Rightarrow \begin{cases} v_1 = 1 \\ v_2 = 1 \end{cases}$$

Отже, знайдений вектор запуску переходів $v=(1,1)$, компоненти якого невід’ємні цілі числа. Це означає, що маркірування $M'=(0,0,0,1)$ може бути досягнуте послідовністю, в якій перехід T_1 присутній один раз і перехід T_2 - один раз. Проте ні одна з послідовностей $T_1 - T_2$ або $T_2 - T_1$ не може бути реалізована в даній сітці Петрі, оскільки в початковому маркіруванні не виконана умова запуску ні переходу T_1 , ні переходу T_2 .

Таким чином, дослідження досяжності матричним способом дає цілком доведену відповідь „недосяжне” тільки у випадку, коли розв’язок рівняння (6.4) або не існує, або серед компонент розв’язку є від’ємні або нецілі числа:

Маркірування M' являється недосяжним, якщо в результаті розв’язання матричного рівняння

$$M' = M + v \cdot D$$

з’ясовано, що розв’язку не існує, або вектор v знайдений та серед його компонент є або нецілі, або від’ємні числа.

В деяких сітках Петрі спостерігається зберігання сумарної кількості фішок в процесі функціонування. Так, наприклад, в сітці, представлений на рисунку 6.3, сумарна кількість фішок в сітці за будь-якого досяжного маркірування, дорівнює 4. Тобто, загальна кількість поламаних та справних пристроїв в розглядуваній системі завжди складає 4. Якщо в сітці Петрі неможливе виникнення і знищення ресурсів, то сітка володіє властивістю зберігання.

Математичне визначення властивості зберігання формулюється так: сітка Петрі являється збережуваною, якщо існує вектор w , $w_i > 0$ для всіх i , такий, що для будь-якого досяжного маркірування M'

$$M \cdot w = M' \cdot w. \quad (6.5)$$

Нехай M' - довільне досяжне маркірування, тоді існує такий вектор запусків переходів v , що:

$$M' = M + v \cdot D. \quad (6.6)$$

Підставимо (6.6) в (6.5):

$$M \cdot w = (M + v \cdot D) \cdot w.$$

Звідси,

$$M \cdot w = M \cdot w + v \cdot D \cdot w.$$

Отже, для довільного вектора v повинно виконуватись

$$0 = v \cdot D \cdot w.$$

Це можливо тільки за умови, що

$$0 = D \cdot w. \quad (6.7)$$

Маємо необхідну і достатню умову збережуваної сітки Петрі:

Сітка Петрі володіє властивістю зберігання тоді і тільки тоді, коли існує вектор w , $w_i > 0$ для всіх i , такий, що

$$D \cdot w = 0,$$

де D – матриця змінювань.

Приклад 4. Перевіримо, чи володіє сітка Петрі, представлена на рисунку 6.3 властивістю зберігання. Пронумеруємо переходи та позиції (рис.6.5):

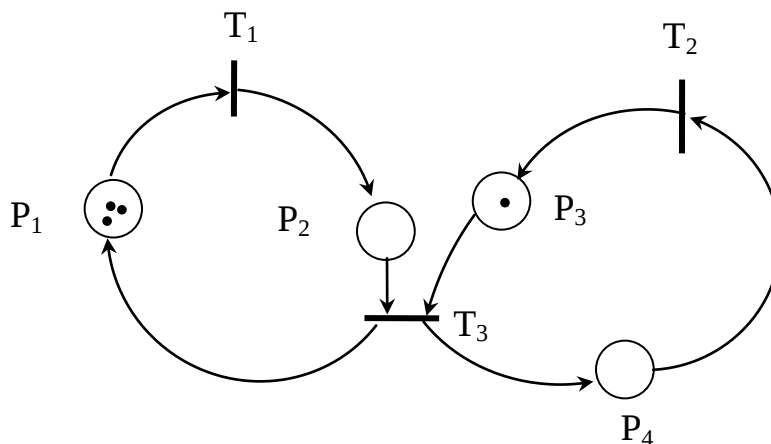


Рисунок 6.5 Формалізована сітка Петрі, яка представляє процес виникнення та усунення несправностей

Складемо матрицю змінювань:

$$D = D^+ - D^- = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix} - \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix} = \begin{pmatrix} -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 1 & -1 & -1 & 1 \end{pmatrix}.$$

Розв'яжемо систему рівнянь для відшукування вектора w , який задовольняє умові (6.6):

$$D \cdot w = 0 \Rightarrow \begin{pmatrix} -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 1 & -1 & -1 & 1 \end{pmatrix} \cdot \begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{pmatrix} = 0 \Rightarrow \begin{cases} -w_1 + w_2 = 0 \\ -w_3 + w_4 = 0 \\ w_1 - w_2 - w_3 + w_4 = 0 \end{cases} \Rightarrow$$

$$\Rightarrow \begin{cases} w_2 = w_1 \\ w_4 = w_3 \\ 0 = 0 \end{cases} \Rightarrow w = (1 \quad 1 \quad 1 \quad 1)$$

Значить сітка Петрі являється зберезуваною.

Приклад 5. Перевіримо, чи володіє сітка Петрі, представлена на рисунку 6.2 властивістю зберігання.

$$D \cdot w = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -1 & -1 & 1 & 0 \\ 0 & 1 & -1 & 1 \\ 0 & 0 & 0 & -1 \end{pmatrix} \cdot \begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{pmatrix} = 0 \Rightarrow \begin{cases} w_1 = 0 \\ -w_1 - w_2 + w_3 = 0 \\ -w_4 = 0 \end{cases} \Rightarrow w = (0 \quad 1 \quad -1 \quad 0)$$

Оскільки серед компонентів вектора w є нульові, сітка Петрі не є зберезуваною.

Зауваження. Зрозуміло, що зберезувана сітка Петрі являється k -обмеженою. І навпаки, якщо не існує такого k , що сітка Петрі являється k -обмеженою, то сітка Петрі не являється зберезуваною. так, для сітки Петрі, представленої на рисунку 6.2, зрозуміло, що в позиції P_2 можливе накопичення фішок до нескінченності. Тому, можна без проведення дослідження стверджувати, що така сітка Петрі не володіє властивістю зберігання.

Якщо з будь-якого досяжного початкового стану можливий перехід в будь-який інший досяжний стан, то сітка Петрі володіє властивістю **живучості**. При дослідженні живучості сітки Петрі вивчають рівні активності переходів.

Перехід T_j має **активність рівня 0**, якщо він ніколи не може бути запущений.

Перехід T_j має **активність рівня 1**, якщо існує маркірування (досягне з початкового), яке дозволяє запуск переходу T_j .

Перехід T_j має **активність рівня 2**, якщо для довільного цілого числа n існує послідовність запусків переходів, в якій перехід T_j присутній принаймні n раз.

Перехід T_j має **активність рівня 3**, якщо існує нескінченна послідовність запусків переходів, в якій перехід T_j присутній необмежено багато разів.

Перехід T_j має **активність рівня 4**, якщо для довільного маркірування M , що є досяжним з початкового маркірування, існує послідовність запусків переходів, яка призводить до маркірування, що дозволяє запуск переходу T_j .

Дослідження активності переходів пояснимо на прикладі (рис. 6.6).

Приклад 6.

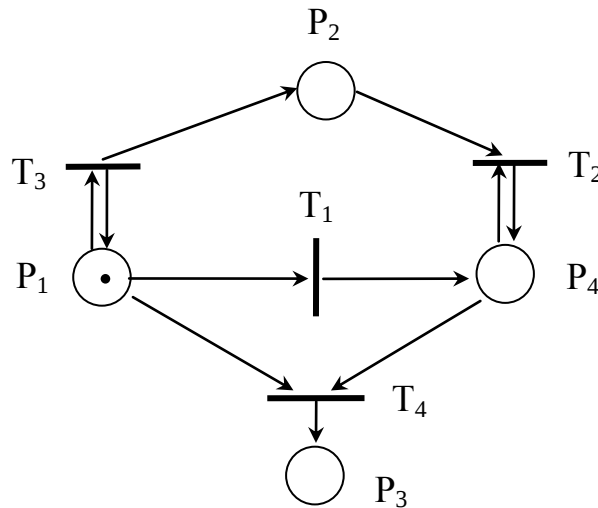


Рисунок 6.6 Сітка Петрі до прикладу 6

Перехід T_1 може бути запущений тільки один раз, після цього фішка з позиції P_1 зникає назавжди. Отже, перехід T_1 має рівень активності 1.

Оскільки фішка в позиції P_4 може з'явитись тільки в результаті запуску переходу T_1 , а запуск цього переходу вимагає вилучення фішки з позиції P_1 , то маркірування, при якому фішка знаходиться і в позиції P_1 , і в позиції P_4 не виникне ніколи. Значить умова запуску переходу T_4 не може бути виконана ніколи. Отже, перехід T_1 має активність 0.

Перехід T_3 може бути запущений нескінченну кількість разів, але як тільки буде запущений перехід T_1 перехід T_3 вже ніколи не запуститься. Тому перехід T_1 активність рівня 3, але не має активність рівня 4.

Перехід T_2 може бути запущений будь-яку кількість разів, але не може бути запущений нескінченну кількість разів. Дійсно, для того, щоб запустити перехід T_2 n разів, потрібно спочатку запустити перехід T_3 n раз, потім запустити перехід T_1 один раз і тільки після цього будуть виконані умови запуску переходу T_2 n разів. Як тільки n накопичених у позиції P_1 будуть вичерпані, умова запуску переходу T_2 зникає назавжди. Тому перехід T_2 активність рівня 2, але не має активність рівня 3.

Таким чином, дослідження сіток Петрі матричним способом дозволяє перевіряти умову запуску переходу та знаходити результат запуску переходу, знаходити результат запуску послідовності переходів, з'ясовувати, чи є маркірування недосяжним та чи володіє сітка Петрі властивістю зберігання.

2. Завдання до роботи

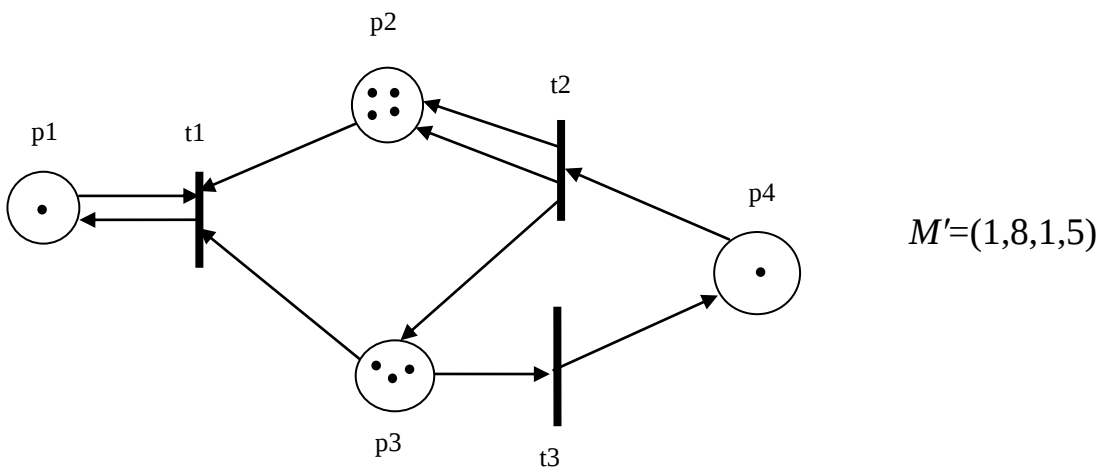
Для даної сітки Петрі

- 1) скласти матрицю змінювань i , розв'язавши матричне рівняння $D \cdot w = 0$, з'ясувати, чи є володіє сітка Петрі властивістю зберігання;
- 2) для даного маркірування M дослідити, чи являється воно досяжним;
- 3) скласти програму, що реалізує запуск N переходів даної сітки Петрі.

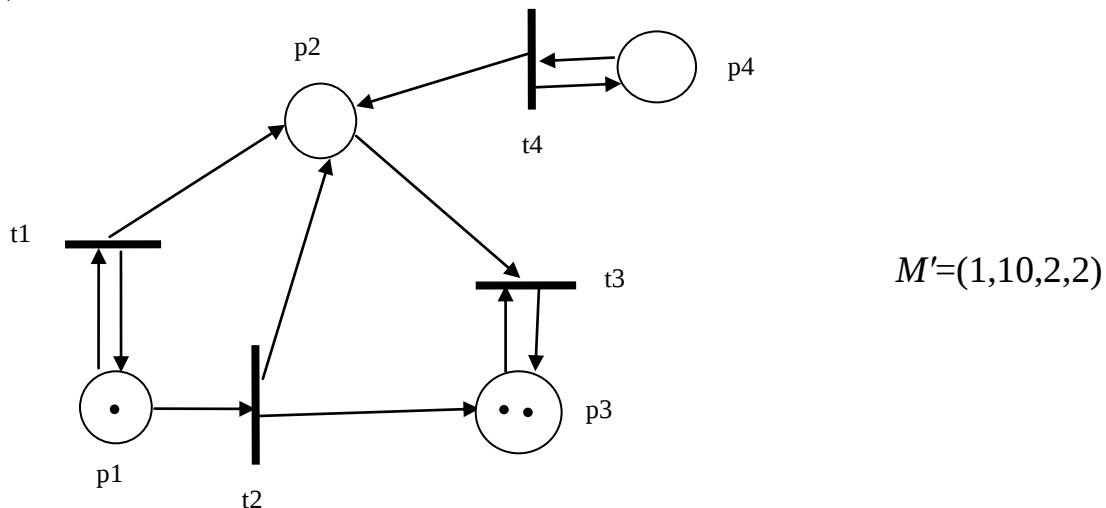
При побудові алгоритму передбачити можливість зміни початкового маркірування.

3. Варіанти

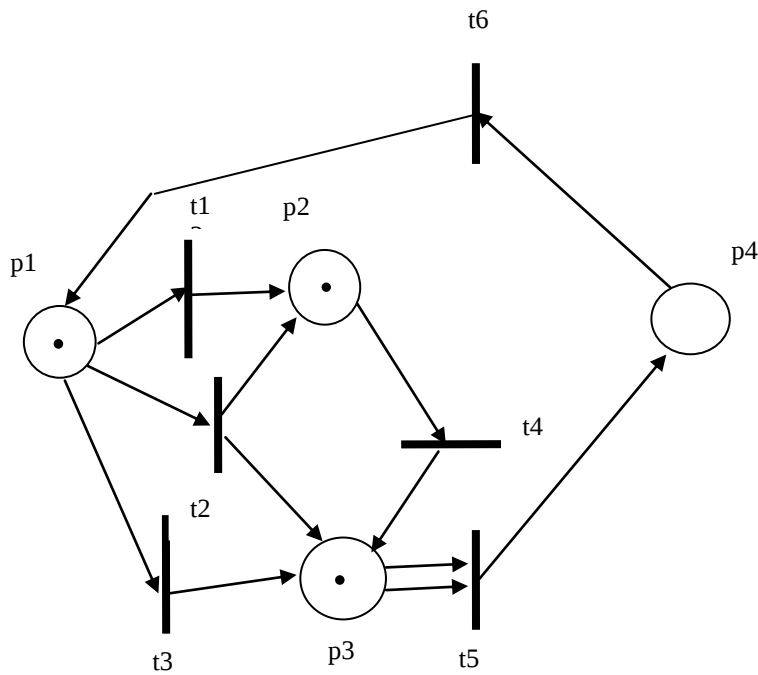
1)



2)

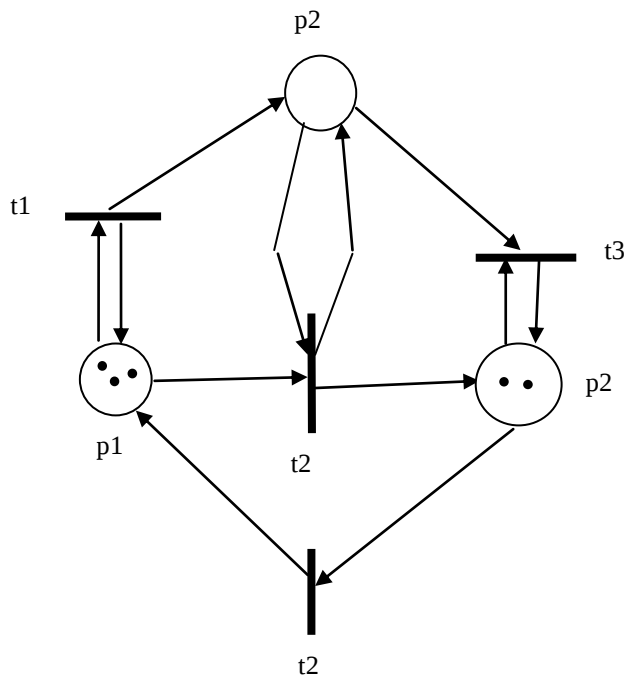


3)



$$M'=(3,0,2,2)$$

4)



$$M'=(0,0,5)$$

4. Хід виконання роботи

- 1) уважно розібрати теоретичні відомості щодо дослідження сіток Петрі матричним способом;
- 2) разом з викладачем вибрати варіант завдання;
- 3) скласти матрицю змінювань;
- 4) розв'язати рівняння (6.6) і зробити висновок, чи володіє дана сітка Петрі властивістю зберігання;
- 5) розв'язати рівняння (6.4) і зробити висновок про досяжність маркірування M' ;

- 6) скласти алгоритм реалізації сітки Петрі використовуючи той засіб програмування, який на Вашу думку найбільш зручний для цієї мети;
- 7) проаналізувати результати і зробити висновки щодо правильності отриманих розрахунків та реалізованого алгоритму;
- 8) оформити звіт.

5. Зміст звіту

- 1) завдання до роботи;
- 2) дослідження збережуваності сітки Петрі;
- 3) дослідження досяжності заданого маркірування;
- 4) лістинг програми;
- 5) результат запуску програми;
- 6) висновки.

6. Контрольні питання

- 1) З яких елементів складається сітка Петрі?
- 2) У чому полягає умова запуску переходу?
- 3) Як здійснюється запуск переходу?
- 4) Як складається матриця входів? матриця виходів? матриця змінювань?
- 5) Які властивості сіток Петрі Ви знаєте?
- 6) Як виконується дослідження збережуваності сітки Петрі?
- 7) Як виконується дослідження досяжності маркірування сітки Петрі?
- 8) Як визначається рівень активності переходу?
- 9) У чому полягає обмеженість застосування матричного підходу до дослідження властивостей сіток Петрі?
- 10) Які труднощі виникли у Вас при реалізації сітки Петрі на ЕОМ? Як Ви їх подолали?
- 11) Чи піддається алгоритм, який Ви склали універсалізації?

7. Література

- 1. Лескин А.А. и др., Сети Петри в моделировании и управлении, 1989.
- 2. Дж. Питерсон, Теория сетей Петри и моделирование систем, М., 1984.

ЛАБОРАТОРНА РОБОТА 7

“Дослідження властивостей сіток Петрі за допомогою дерева досяжності”

Мета роботи: навчитися складати дерево досяжності сітки Петрі та виконувати дослідження властивостей сіток Петрі.

1. Короткі теоретичні відомості з сіткового планування робіт

Дерево досяжності представляє множину досяжних маркірувань сітки Петрі. Дерево досяжності розпочинається з початкового маркірування, а закінчується термінальним або дублюючим маркіруванням. Термінальним маркіруванням називається маркірування, в якому жоден з переходів сітки

Петрі не запускається. Назва термінальне походить від англійського слова *terminate*, що в перекладі означає обрубувати. Дублюючим маркіруванням називається маркірування, що раніше зустрічалося в дереві досяжності. Якщо в деякій позиції сітки Петрі спостерігається зростання кількості фішок до нескінченності, то для зображення цього факту використовують символ ω в цій позиції. Символ ω в позиції M_j маркірування M з'являється тоді, коли на путі до маркірування M спостерігалось маркірування M' , в якому всі значення, крім j -ого співпадають зі значеннями маркірування M , а значення M'_j перевищує значення M_j :

$$\begin{aligned} M'_k &= M_k, \quad k \neq j \\ M'_j &> M_j \end{aligned}$$

Одного разу з'явившись, символ ω уже не змінюється і не зникає в дереві досяжності: додавання або віднімання від нескінченності є нескінченність.

Приклад 1. Побудуємо дерево досяжності для сітки, представленій на рисунку 7.1.

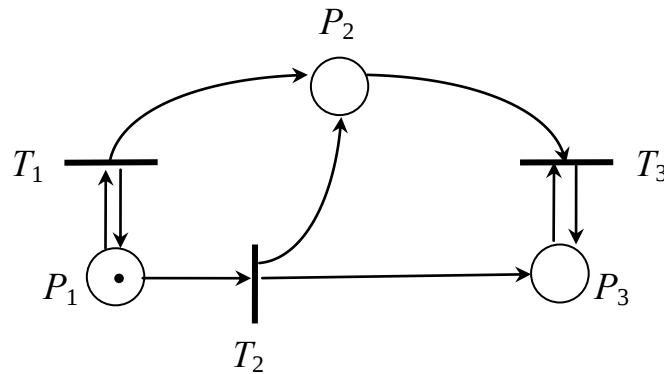


Рисунок 7.1 Сітка Петрі до прикладу 1

Початкове маркірування $M=(1,0,0)$ дозволяє запуск переходів T_1 та T_2 (рис. 7.2). Запуск переходу T_1 призводить до маркірування $M=(1,1,0)$, компоненти якого всі, крім другої, співпадають з компонентами початкового маркірування. Крім того, в другій компоненті спостерігається зростання кількості фішок. Отже, в цьому місці з'являється символ ω . Маркірування $M=(1,\omega,0)$ знову дозволяє запуск переходів T_1 та T_2 , що приводить до дублюючого маркірувань $M=(1,\omega,0)$ та маркірування $M=(0,\omega,1)$. Маркірування $M=(0,\omega,1)$ дозволяє запуск тільки переходу T_3 , що призводить до дублюючого маркірування $M=(0,\omega,1)$.

Запуск переходу T_2 з початкового маркірування призводить до маркірування $M=(0,1,1)$, яке дозволяє тільки запуск переходу T_3 . В результаті запуску переходу T_3 виникає маркірування $M=(0,0,1)$, яке є термінальним.

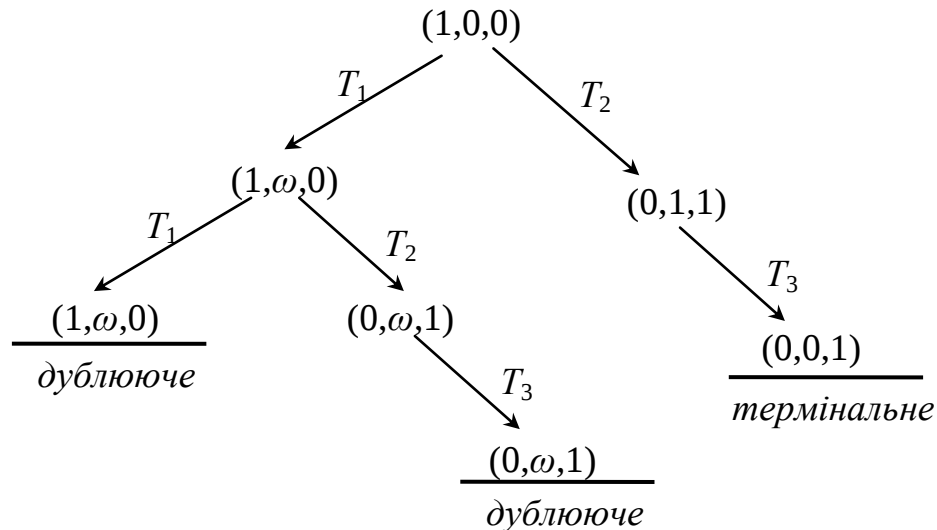


Рисунок 7.2 Дерево досяжності сітки Петрі, представленої на рисунку 7.1.

Викладемо **алгоритм побудови дерева досяжності**. Кожне маркірування дерева досяжності класифікується як початкове, граничне (те, що підлягає обробці), термінальне, дублююче та внутрішнє. Внутрішнім є маркірування дерева досяжності, що не є початковим, дублюючим або термінальним. Обробка M – граничне маркірування здійснюється за наступними правилами:

1. Якщо в дереві досяжності є інше маркірування M' , що не являється граничним, таке, що $M'=M$, то маркірування M – дублююче.
2. Якщо для маркірування M жоден з переходів сітки Петрі не запускається, то маркірування M – термінальне.

3. Для всякого дозволеного переходу T_n створити маркірування X дерева досяжності та дугу, що йде від M до X , і відмітити її символом T_n . Об'явити маркірування M – внутрішнім, а маркірування X – граничним. Значення X_j маркірування X визначити таким чином:

- 1) якщо в позиції j маркірування M знаходиться символ ω , то в позиції j маркірування X теж поставити символ ω :

$$M_j = \omega \Rightarrow X_j = \omega;$$

- 2) якщо на путі від початкового маркірування існує маркірування M' таке, що $M'_k \leq X_k$, $k \neq j$, $M'_j < X_j$, то поставити в позиції j маркірування X символ ω :

$$M'_k \leq X_k, \quad k \neq j, \quad M'_j < X_j \Rightarrow X_j = \omega;$$

- 3) інакше присвоїти позиції j маркірування X результат запуску переходу T_n :

$$X_j = M_j + D_{nj}.$$

Алгоритм побудови дерева досяжності закінчується, коли всі маркірування його термінальні, дублюючі або внутрішні.

Приклад 2. Розглянемо наступну сітку Петрі (рис. 7.3)

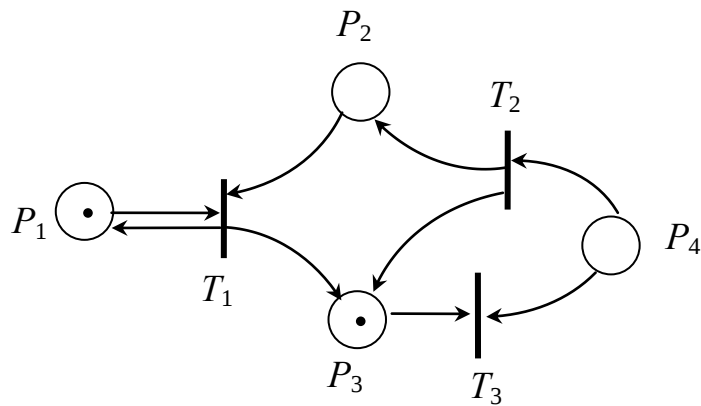


Рисунок 7.3 Сітка Петрі до прикладу 2

Побудуємо дерево досяжності (рис. 7.4). В початковому маркіруванні дозволено є тільки запуск переходу T_3 . Запуск цього переходу приводить до маркірування $M=(1,0,0,1)$, яке дозволяє запуск тільки переходу T_2 . Результатом запуску переходу T_2 є маркірування $M=(1,1,1,0)$. Порівнюючи це маркірування з маркіруванням $M=(1,0,1,0)$, яке є на шляху до розглядуваного, приходимо до висновку, що в позиції M_2 спостерігається зростання кількості фішок. Тому в цій позиції з'являється символ нескінченності ω :

$$(1,1,1,0) \geq (1,0,1,0) \Rightarrow M' \geq M \Rightarrow M' = (1, \omega, 1, 0).$$

Маркірування $(1, \omega, 1, 0)$ дозволяє запуск переходів T_1 та T_3 . Запуск переходу T_1 призводить до маркірування $(1, \omega, 0, 0)$, яке є термінальним. А запуск переходу T_3 - до маркірування $(1, \omega, 0, 1)$, яке дозволяє запуск тільки переходу T_3 .

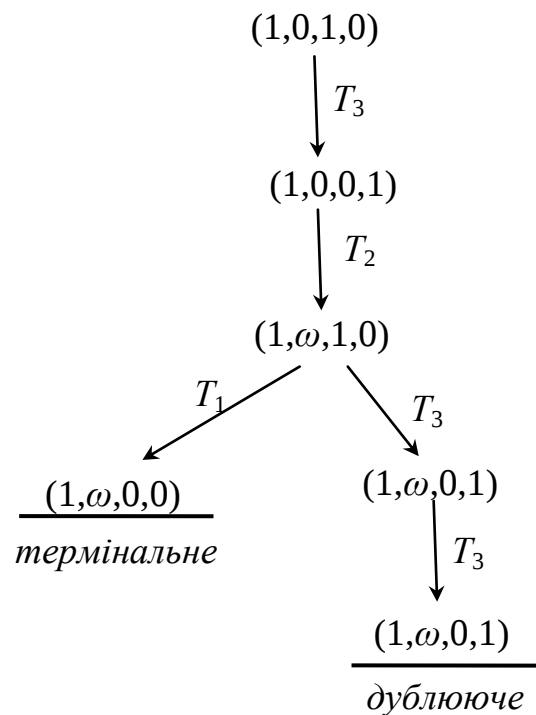


Рисунок 7.4 Дерево досяжності сітки Петрі, представленої на рисунку 7.3.

Побудоване дерево досяжності використовують для дослідження властивостей сітки Петрі.

Сітка Петрі k -обмежена тоді і тільки тоді, коли в її дереві досяжності відсутній символ ω . Розташування символу ω вказує, які позиції сітки необмежені. Число k визначається найбільшим значенням кількості фішок, яке присутнє у дереві досяжності.

Якщо сітка Петрі k -обмежена, можливо, вона володіє властивістю зберігання. Щоб дослідити, чи є сітка Петрі збережуваною, потрібно для кожного маркірування, яке присутнє у дереві досяжності, скласти систему рівнянь

$$\begin{cases} w_1 \cdot M_1^{(1)} + w_2 \cdot M_i^{(1)} + \dots + w_p \cdot M_p^{(1)} = s \\ \cdot \\ \cdot \\ \cdot \\ w_1 \cdot M_1^{(m)} + w_2 \cdot M_i^{(m)} + \dots + w_p \cdot M_p^{(m)} = s \end{cases}, \quad (7.1)$$

де p – кількість позицій сітки Петрі, m – кількість маркірувань в дереві досяжності сітки Петрі, s – невідома величина.

Якщо система рівнянь (7.1) має розв'язок, усі значення якого цілі невід'ємні числа:

$$s > 0, s \in \mathbb{Z}, w_i > 0, w_i \in \mathbb{Z}, i = 1, \dots, p, \quad (7.2)$$

то сітка Петрі володіє властивістю зберігання.

Приклад 3. Розглянемо наступну сітку Петрі (рис. 7.5)

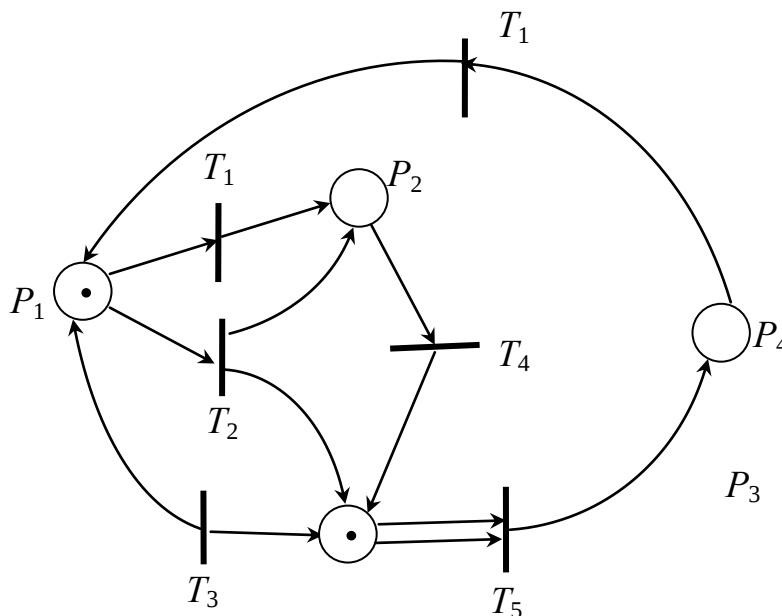


Рисунок 7.5 Сітка Петрі до прикладу 3

Побудуємо дерево досяжності (рис. 7.6). В початковому маркіруванні дозволим є тільки запуск переходу T_3 . Запуск цього переходу приводить до маркірування $M=(1,0,0,1)$, яке дозволяє запуск тільки переходу T_2 . Результатом

запуск переходу T_2 є маркірування $M=(1,1,1,0)$. Порівнюючи це маркірування з маркіруванням $M=(1,0,1,0)$, яке є на шляху до розглядуваного, приходимо до висновку, що в позиції M_2 спостерігається зростання кількості фішок. Тому в цій позиції з'являється символ нескінченності ω :

$$(1,1,1,0) \geq (1,0,1,0) \Rightarrow M' \geq M \Rightarrow M' = (1, \omega, 1, 0).$$

Маркірування $(1, \omega, 1, 0)$ дозволяє запуск переходів T_1 та T_3 . Запуск переходу T_1 призводить до маркірування $(1, \omega, 0, 0)$, яке є термінальним. А запуск переходу T_3 - до маркірування $(1, \omega, 0, 1)$, яке дозволяє запуск тільки переходу T_3 .

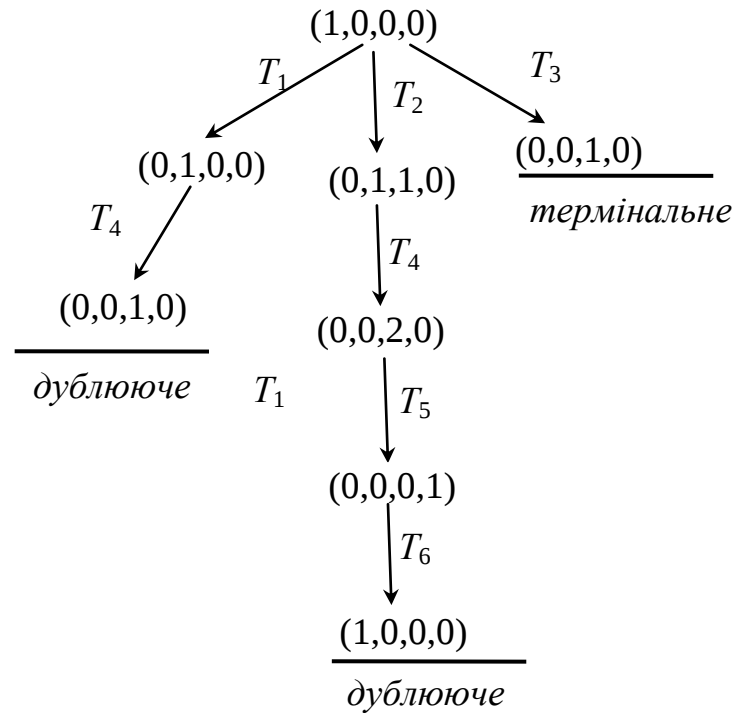


Рисунок 7.6 Дерево досяжності сітки Петрі, представленої на рисунку 7.5.

В дереві досяжності символ ω відсутній, найбільше спостережуване значення кількості фішок дорівнює 2. тому сітка Петрі є 2-обмеженою. З'ясуємо, чи є вона зберігаючою. Складемо рівняння (7.1):

$$\begin{cases} w_1 \cdot 1 + w_2 \cdot 0 + w_3 \cdot 0 + w_4 \cdot 0 = s \\ w_1 \cdot 0 + w_2 \cdot 1 + w_3 \cdot 0 + w_4 \cdot 0 = s \\ w_1 \cdot 0 + w_2 \cdot 0 + w_3 \cdot 1 + w_4 \cdot 0 = s \\ w_1 \cdot 0 + w_2 \cdot 1 + w_3 \cdot 1 + w_4 \cdot 0 = s. \\ w_1 \cdot 0 + w_2 \cdot 0 + w_3 \cdot 2 + w_4 \cdot 0 = s \\ w_1 \cdot 0 + w_2 \cdot 0 + w_3 \cdot 0 + w_4 \cdot 1 = s \end{cases} \Rightarrow \begin{cases} w_1 = s \\ w_2 = s \\ w_3 = s \\ w_2 + w_3 = s \\ 2w_3 = s \\ w_4 = s \end{cases} \Rightarrow \begin{cases} w_1 = s \\ w_2 = s \\ w_3 = s \\ 2s = s \\ 2s = s \\ w_4 = s \end{cases} \Rightarrow \begin{cases} w_1 = 0 \\ w_2 = 0 \\ w_3 = 0 \\ s = 0 \\ w_4 = 0 \end{cases}$$

Оскільки компоненти розв'язку не задовольняють умові (7.2), то сітка Петрі не являється зберігаючою.

Якщо маркірування M' присутнє в дереві досяжності сітки Петрі, то воно являється досяжним. Якщо в дереві досяжності присутнє маркірування M'' таке, що $M'' \geq M'$, то маркірування M' *покривається*. Наприклад, маркірування $(1, \omega, 1, 0)$ покриває маркірування $(1, 15, 1, 0)$.

Приклад 4. Розглянемо наступну сітку Петрі (рис. 7.7) на досяжність маркірування $M=(0,14,1,7)$.

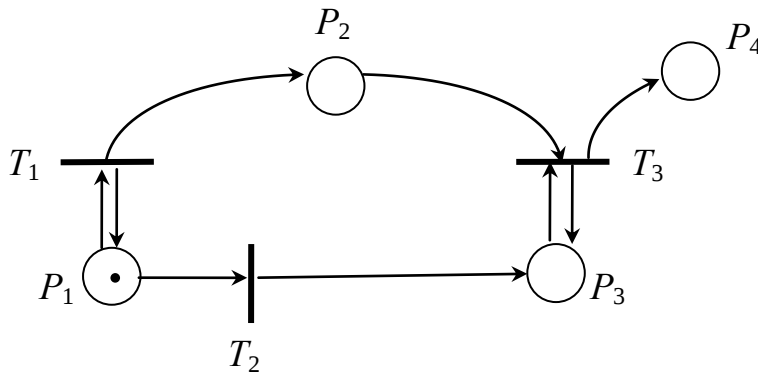


Рисунок 7.7 Сітка Петрі до прикладу 4

Побудуємо дерево досяжності (рис. 7.8).

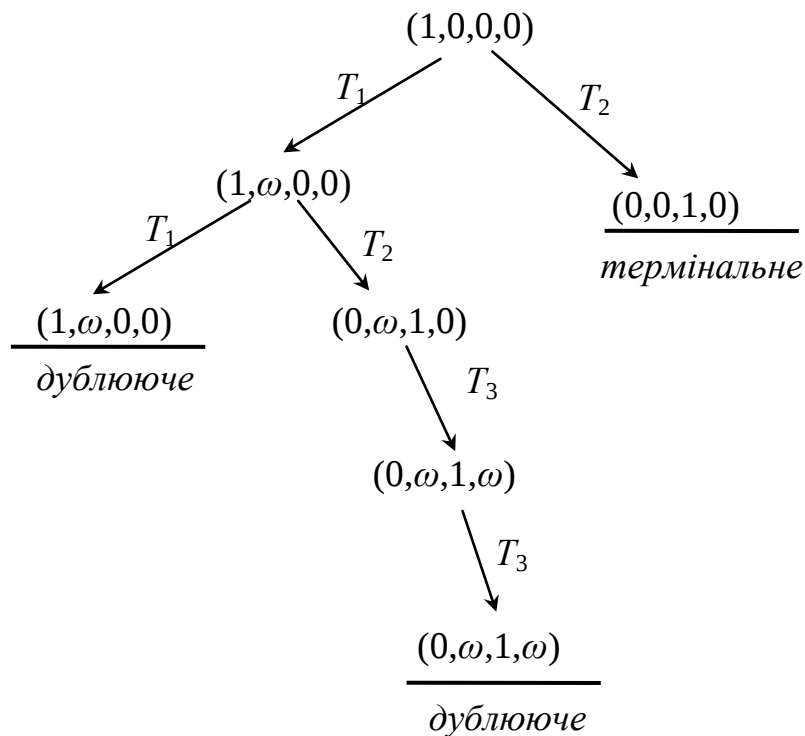


Рисунок 7.8 Дерево досяжності сітки Петрі, представленої на рисунку 7.7.

Оскільки в дереві досяжності присутнє маркірування $M'=(0,\omega,1,\omega)$, то маркірування $M=(0,14,1,7)$ покривається сіткою Петрі. Крім того, з дерева досяжності видно, що маркірування M' досягається в результаті кількох запусків переходу T_1 , потім декількох запусків переходу T_2 та декількох запусків переходу T_3 . З огляду на сітку Петрі, можна побудувати послідовність переходів, що забезпечує досяжність маркірування $M=(0,14,1,7)$:

$$\underbrace{T_1 - \dots - T_1}_{21} - T_2 - \underbrace{T_3 - \dots - T_3}_7.$$

Тобто перехід T потрібно запустити 21 раз і результатом цього буде маркірування $(1,21,0,0)$. Потім запустити перехід T і отримати маркірування

(0,21,1,0). Нарешті запустити 7 раз перехід T і отримати маркірування (0,14,1,7).

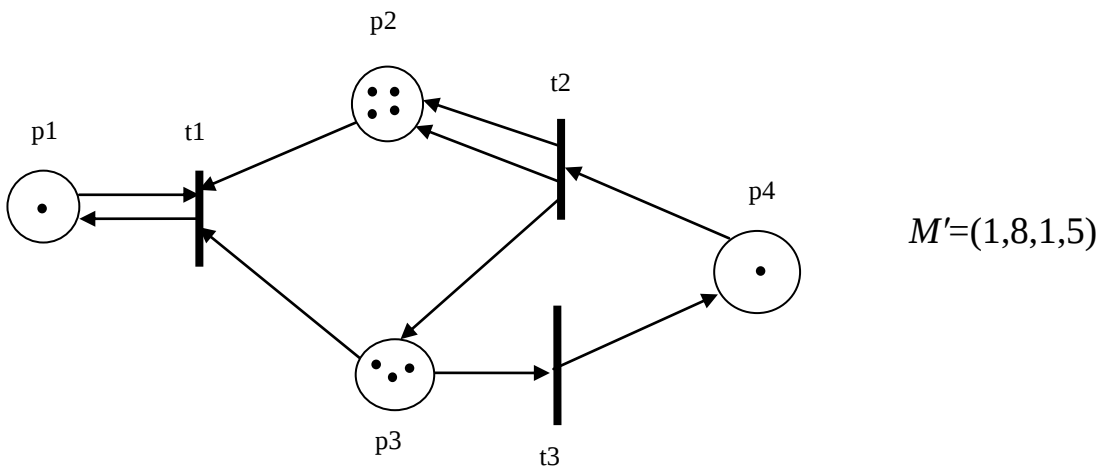
Таким чином, за допомогою дерева досяжності досліджуються такі властивості сітки Петрі як k -обмеженість, збережуваність та покриття маркірування. Досяжність маркірування не може бути досліджена досконало, оскільки з дерева досяжності залишається невідомою послідовність переходів, що призводить до заданого маркірування.

2. Завдання до роботи

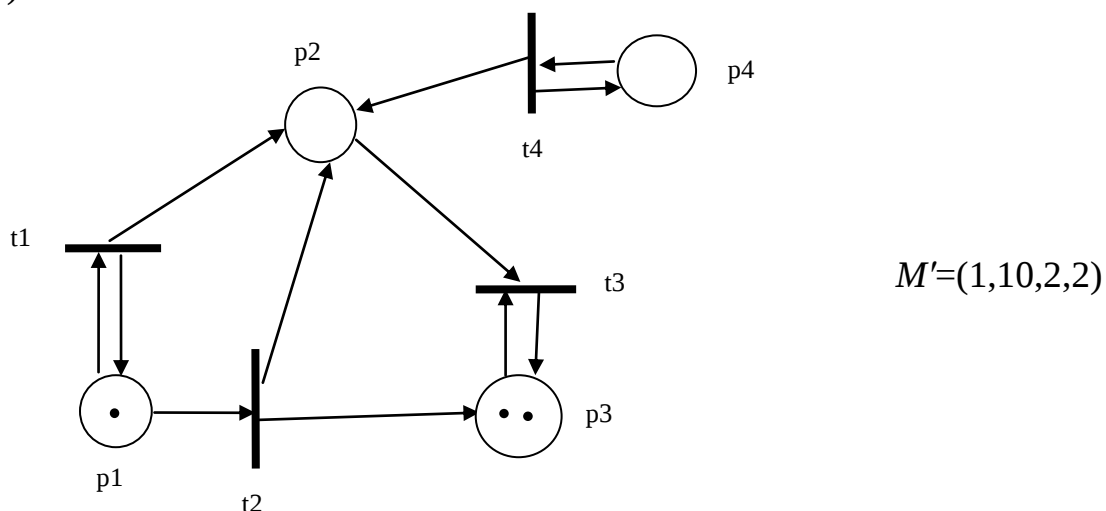
Для сітки Петрі, що була досліджена в лабораторній роботі 6 скласти дерево досяжності маркірувань та провести дослідження властивостей сітки Петрі. При побудові алгоритму передбачити можливість зміни початкового маркірування.

3. Варіанти

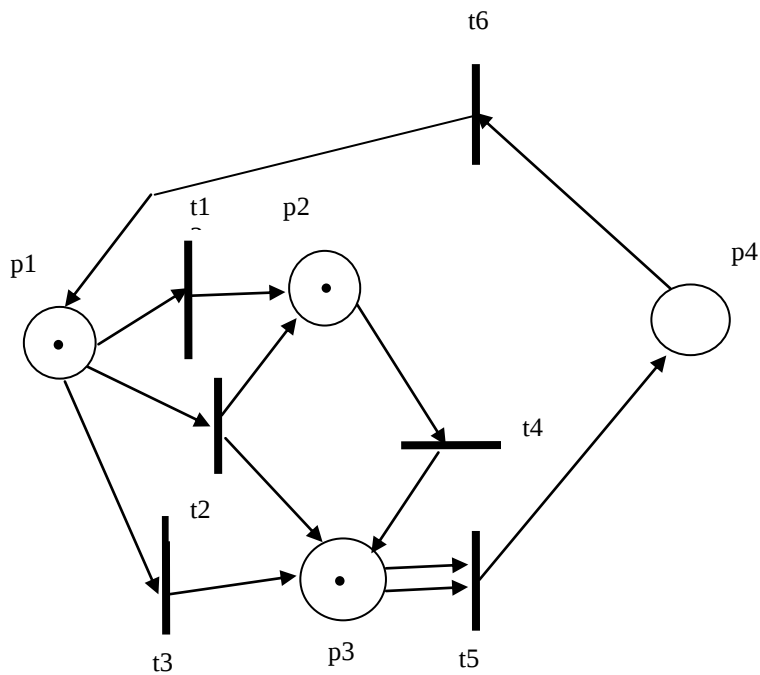
1)



2)

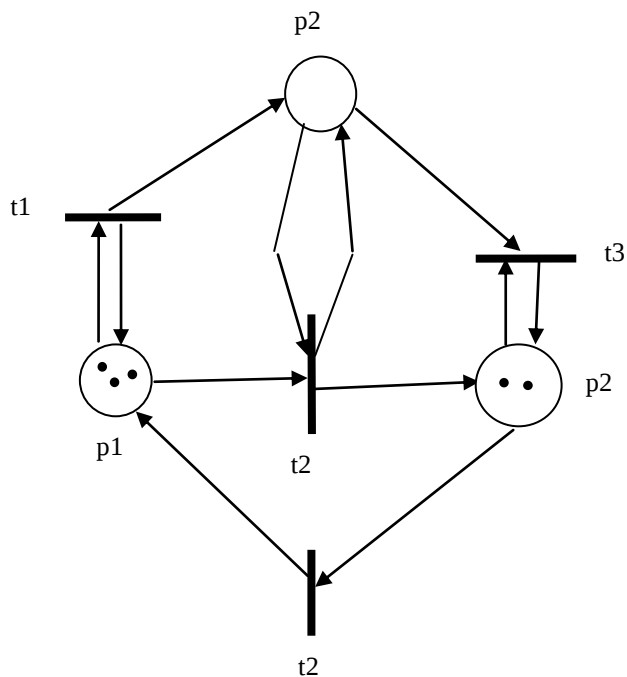


3)



$$M'=(3,0,2,2)$$

4)



$$M'=(0,0,5)$$

4. Хід виконання роботи

- 1) уважно розібрати теоретичні відомості щодо дослідження сіток Петрі за допомогою дерева досяжності;
- 2) разом з викладачем вибрати варіант завдання;
- 3) скласти алгоритм побудови дерева досяжності сітки Петрі використовуючи той засіб програмування, який на Вашу думку найбільш зручний для цієї мети;

- 4) з'ясувати досяжність маркірування M' ;
- 5) з'ясувати збережуваність сітки Петрі;
- 6) проаналізувати результати і зробити висновки щодо правильності отриманих розрахунків;
- 7) оформити звіт.

5. Зміст звіту

- 1) завдання до роботи;
- 2) дерево досяжності сітки Петрі;
- 3) дослідження збережуваності сітки Петрі;
- 4) дослідження досяжності заданого маркірування;
- 5) лістинг програми;
- 6) результат запуску програми;
- 7) висновки.

6. Контрольні питання

- 1) Що називають деревом досяжності сітки Петрі?
- 2) Як складається дерево досяжності сітки Петрі?
- 3) Опишіть алгоритм складання дерева досяжності.
- 4) Які властивості сітки Петрі можна дослідити за допомогою дерева досяжності?
- 5) Як виконується дослідження збережуваності сітки Петрі?
- 6) Як виконується дослідження досяжності маркірування сітки Петрі?
- 7) Що розуміють під покриттям маркірування?
- 8) У чому полягає обмеженість застосування дерева досяжності до дослідження властивостей сіток Петрі?
- 9) Які труднощі виникли у Вас при складанні дерева досяжності на ЕОМ? Як Ви їх подолали?
- 10) Чи піддається алгоритм, який Ви склали універсалізації?

7. Література

1. Лескин А.А. и др., Сети Петри в моделировании и управлении, 1989.
2. Дж. Питерсон, Теория сетей Петри и моделирование систем, М., 1984.