ФАКУЛЬТЕТ УПРАВЛІННЯ ПРОЦЕСАМИ ПЕРЕВЕЗЕНЬ

Кафедра „Вища математика”

МАТЕМАТИЧНЕ ПРОГРАМУВАННЯ

МЕТОДИЧНІ ВКАЗІВКИ

та завдання до виконання контрольних робіт

Частина 2

Харків – 2009

Методичні вказівки розглянуто та рекомендовано до друку на засіданні кафедри вищої математики 28 січня 2008 р.,

протокол № 8.

Призначені для студентів заочної форми навчання ФЕТ та відповідають робочій програмі з курсу "Математичне програмування".

Укладачі:

доценти О.О. Думіна,
М.Є. Резуненко,
О.І. Удодова,
старш. викл. Ю.С. Шувалова

Рецензент

доц. Н.С. Юрчак

МАТЕМАТИЧНЕ ПРОГРАМУВАННЯ

Методичні вказівки та завдання до виконання контрольних робіт

Частина 2

Відповідальний за випуск Шувалова Ю.С.

Редактор Еткало О.О.

Підписано до друку 31.03.08 р.
Формат паперу 60х84 1/16 . Папір писальний.
Умовн.-друк.арк. 3,0. Обл.-вид.арк. 3,25.
Замовлення № Тираж 300 Ціна

Видавництво УкрДАЗТу, свідоцтво ДК 2874 від 12.06.2007 р.
Друкарня УкрДАЗТу,
61050, Харків - 50, пл. Фейєрбаха, 7

УКРАЇНСЬКА ДЕРЖАВНА АКАДЕМІЯ
ЗАЛІЗНИЧНОГО ТРАНСПОРТУ

Кафедра “Вища математика”

МАТЕМАТИЧНЕ ПРОГРАМУВАННЯ

Частина II

МЕТОДИЧНІ ВКАЗІВКИ ТА ЗАВДАННЯ

**до виконання контрольних робіт
для студентів ФЕТ заочної форми навчання**

Харків 2009

Методичні вказівки розглянуто та рекомендовано до друку на засіданні кафедри вищої математики 28 січня 2008 р., протокол № 8.

Призначені для студентів заочної форми навчання ФЕТ та відповідають робочій програмі з курсу “Математичне програмування”.

Укладачі:

доценти О.О. Думіна,
М.Є. Резуненко,
О.І. Удодова,
старш. викл. Ю.С. Шувалова

Рецензент

доц. Н.С. Юрчак

ВСТУП

Методичні вказівки містять теоретичні відомості з основних розділів математичного програмування, зразки розв'язання задач з розгорнутими поясненнями, варіанти завдань для виконання контрольної роботи, список літератури.

Методичні вказівки рекомендовані для студентів заочної форми навчання факультету економіки транспорту, але можуть бути використані і при вивченні цього розділу студентами стаціонару.

Номери варіантів індивідуальних завдань видаються викладачем. Залік контрольних робіт згідно з навчальною програмою є необхідною умовою допуску студента до заліку або екзамену з дисципліни „Математичне програмування”. Контрольна робота, що містить виконаний чужий варіант завдань, не зараховується.

1 ЕЛЕМЕНТИ ТЕОРІЇ ІГОР

Предметом теорії ігор є зіткнення інтересів різних людей, протиріччя між цими інтересами. В економіці таким зіткненням є конкуренція, у військовій справі – організація тилу, протиповітряна оборона, дуель двох супротивників тощо. Математичні успіхи теорії ігор у теперішній час достатньо обмежені. Вона не пропонує поведінку, що веде до виграшу, а лише вказує, чого може досягти гравець у найгіршій для нього ситуації і як він повинен діяти, щоб отримати мінімальний програш (або максимальний виграш). Але і це, безумовно, є корисним. А рекомендації щодо виграшу – справа майбутньої теорії ігор.

Будь-яка гра передбачує таке:

1) наявність деякої кількості *гравців* – осіб, що беруть участь у грі. Можуть бути ігри з одним гравцем (пасьянс), двома гравцями (шахи, дві конкуруючі фірми), трьома гравцями (преферанс, три фірми на ринку) тощо. За кількістю гравців іде класифікація ігор – *ігри двох осіб*, *трьох* тощо;

2) кінцевий *виграш* (*програш*) кожного гравця. Коли гра закінчується, кожний гравець отримує доход d_i (якщо $d_i < 0$ – гравець програв), який залежить від його поведінки та поведінки інших гравців.

Найбільш вивченим класом ігор є так звані *ігри з нульовою сумою*, коли в будь-якій *партії* наявна умова $\sum_{i=1}^{n} d_i = 0$, тобто якщо хтось виграє, то хтось обов'язково програє. Наприклад, в *іграх двох осіб з нульовою сумою* $d_1 + d_2 = 0$, тобто $d_2 = -d_1$. В цьому випадку інтереси гравців строго протилежні, бо виграш одного є одночасно програшем іншого. Такі ігри називаються *антагоністичними*.

Будь-яка гра складається з *партій*, які починаються та закінчуються, після чого гравцям сплачуються їхні виграші. У свою чергу кожна партія складається з *ходів*, які одночасно або послідовно роблять гравці. Опис гри як послідовності ходів має назву *позиційної форми гри* – теорія таких ігор розроблена ще дуже слабко. Основний зміст сучасної теорії ігор – це так звана *матрична форма гри*. У цьому випадку вважається, що кожний гравець робить лише один хід, причому усі гравці ходять одночасно. Після цього кожному гравцю сплачується виграш (або береться програш) в залежності від того, які ходи були зроблені ним та іншими гравцями. Взагалі гра в позиційній формі може бути зведена до гри в матричній формі, але для реальних ігор це настільки складно, що майже неможливо виконати навіть на сучасних ЕОМ. Однак у майбутньому таке зведення може мати і практичний сенс.

Гра двох осіб з нульовою сумою

Гра двох осіб з нульовою сумою у матричній формі посідає центральне місце в сучасній теорії ігор, бо теорія таких ігор розроблена майже до кінця.

Нехай є два гравця. Перший гравець має n можливих ходів $i = 1,2,3...n$, другий гравець має m можливих ходів $j = 1,2,3...m$. Ці можливі ходи називаються *чистими стратегіями* гравців. Обидва гравця роблять одночасно по одному ходу, після чого партія вважається завершеною. Якщо перший гравець робить хід i, а другий – хід j, то перший отримує виграш a_{ij}. Очевидно, що програш другого гравця складає також a_{ij}. Ці дані можна записати у вигляді матриці

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix},$$

у якій рядки відповідають ходу першого гравця, а стовпчики – ходу другого гравця. Ця матриця називається *платіжною матрицею* гри. Які ж ходи мають робити гравці?

Ігри із сідловою точкою

Розглянемо гру з такою платіжною матрицею:

$$\begin{array}{r c} & \begin{array}{cccc} j=1 & 2 & 3 & 4 \end{array} \\ \begin{array}{r} i=1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} & \begin{pmatrix} 18 & 3 & 0 & 2 \\ 0 & 3 & 8 & 20 \\ 5 & 4 & 5 & 5 \\ 16 & 4 & 2 & 25 \\ 9 & 3 & 0 & 20 \end{pmatrix} \end{array}.$$

Спробуємо поміркувати з точки зору першого гравця. Якщо він зробить хід $i=1$, то найгіршою для нього буде ситуація, коли другий гравець зробить хід $j=3$, бо в цьому випадку перший гравець отримає 0. Якщо перший гравець зробить хід $i=2$, то в найгіршому випадку (при ході другого гравця $j=1$) він також отримає 0. Аналогічно, при $i=3$ він у найгіршому випадку отримає 4 (при $j=2$), при $i=4$ – 2 (при $j=3$) і, нарешті, при $i=5$ він у найгіршому випадку отримає 0 (при $j=3$). Прагнучи зробити свій гарантований виграш найбільшим, перший гравець має обрати хід $i=3$, бо в цьому випадку він гарантує собі виграш, що дорівнює 4 (правда, і максимальний його виграш при цьому невеликий – лише 5).

Тепер спробуємо поглянути на цю матрицю з точки зору другого гравця. Для нього це матриця його програшу. Якщо він вибере хід $j=1$, то його максимальний програш буде дорівнювати 18 (якщо перший гравець зробить хід $i=1$). Аналогічно, при $j=2$ максимальний програш другого гравця буде дорівнювати 4, при $j=3$ – програш 8, і, нарешті, при $j=4$ його максимальний програш буде дорівнювати 25. Прагнучи зробити свій максимальний програш щонайменшим, другий гравець має обрати хід $j=2$, бо в цьому випадку його максимальний програш, що дорівнює 4, є найменшим.

Отже, ми прийшли до висновку, що перший гравець має ходити $i=3$, а другий $j=2$. Припустимо тепер, що другий повідомляє першого, що він буде робити хід $j=2$. І в цьому випадку найкращим ходом для першого гравця залишається $i=3$. Аналогічно, інформація про хід першого гравця $i=3$ не змінить найкращий хід другого гравця $j=2$. Кажуть, що пара $i=3$, $j=2$ є *розв'язком гри*, а величина виграшу при цьому першого гравця (і одночасно величина програшу другого) 4 – це *ціна гри*.

Запишемо це математично. Нехай перший гравець обирає хід i. У найгіршій для нього ситуації він виграє $\min\limits_j a_{ij}$. Прагнучи зробити свій мінімальний виграш найбільшим, він обирає свій хід із умови $\max\limits_i \min\limits_j a_{ij}$. Така стратегія називається *максимінною.* Аналогічно, другий гравець, обираючи хід j, у найгіршій для себе ситуації програє $\max\limits_i a_{ij}$. Прагнучи зробити свій максимальний програш найменшим, він має обирати свій хід із умови $\min\limits_j \max\limits_i a_{ij}$. Така стратегія називається *мінімаксною*.

Теорема. $$\max_i \min_j a_{ij} \le \min_j \max_i a_{ij}.$$

У нашому ж прикладі $\max\limits_i \min\limits_j a_{ij} = \min\limits_j \max\limits_i a_{ij}$. Ця рівність забезпечує наявність розв'язку гри у чистих стратегіях. Елемент матриці, на якому досягається ця рівність (у нашому випадку $a_{32}=4$), є *сідловою точкою* матриці, він є максимальним у своєму стовпчику і мінімальним у своєму рядку.

Виникає декілька питань щодо сідлових точок.

1 Чи може матриця мати декілька сідлових точок? Так, може.

Наприклад, у матриці $\begin{pmatrix} \mathbf{12} & 13 & \mathbf{12} \\ 10 & 31 & 9 \end{pmatrix}$ є дві сідлові точки $a_{11} = a_{13} = 12$.

2 Якщо сідлових точок декілька, то протиріч між ними не виникає, вони завжди дорівнюють одна одній і будь-яка з них дає розв'язок матричної гри та ціну гри.

3 Не кожна матриця має сідлову точку. Наприклад, у матриці $\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$ сідлових точок немає.

Мішані стратегії

Сідлова точка в матричних іграх скоріше виняток, ніж правило. А що може гарантувати собі гравець, якщо сідлової точки немає?

Розглянемо гру з платіжною матрицею $\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$. Тут $\max\limits_i \min\limits_j a_{ij} = -1$, $\min\limits_j \max\limits_i a_{ij} = 1$. Уявимо себе в позиції першого гравця. Він має гарантований виграш (скоріше, програш), що дорівнює -1. Як він може його підвищити? Звичайно, якщо гра повторюється багато разів, можна вивчити поведінку свого партнера, вигадувати якісь схеми тощо. Але це не спрацює, якщо кількість партій невелика. У такій ситуації єдиний вихід – обирати хід випадково, наприклад, підкинувши монету. Що це дасть? Виграш стане випадковою величиною і оцінювати його треба за математичним очікуванням. Нехай другий гравець робить хід $j = 1$. Тоді математичне очікування виграшу першого гравця буде $1 \cdot \frac{1}{2} + (-1) \cdot \frac{1}{2} = 0$. Якщо другий гравець робить хід $j = 2$, то математичне очікування виграшу першого гравця буде $(-1) \cdot \frac{1}{2} + 1 \cdot \frac{1}{2} = 0$. Таким чином, обираючи свій хід випадково, перший гравець гарантує собі (в середньому, а не в кожній партії) виграш, що дорівнює 0, а це все-таки краще, ніж гарантований виграш (-1). Аналогічно, другий гравець, обираючи навмання свій хід, гарантує собі в середньому програш, що дорівнює 0 – це теж краще, ніж програш, що дорівнює 1. Таким чином, виявляється, що випадковий вибір хода підвищує шанси на успіх, хоча б в середньому. І це є однією з основних ідей теорії ігор – **обирати свій хід випадково**. Такий випадковий вибір ходу має назву *мішаної стратегії*.

Оформимо тепер це математично. Розглянемо спочатку ситуацію, коли всі $a_{ij} > 0$ (для виконання цієї умови *можна додати одне і те ж саме число до всіх елементів платіжної матриці, оптимальні стратегії гравців при цьому не зміняться, а ціна гри збільшиться на додану величину*). Це гарантує нам, що виграш першого гравця буде додатним. Нехай у розпорядженні першого гравця є n ходів $i = 1, 2, 3 ... n$ і він обирає їх випадково, так що хід з номером i робиться з імовірністю p_i. Набір чисел $(p_1, p_2, ..., p_n)$ називається *мішаною стратегією* першого гравця. Із властивостей імовірностей випливає, що всі $p_i \geq 0$ та $\sum_{i=1}^{n} p_i = 1$. Нехай v є гарантований середній виграш першого гравця. Це означає, що при будь-якому ході другого гравця перший гравець отримає <u>в</u> середньому не менше, ніж v, тобто

$$\begin{cases} a_{11}p_1 + a_{21}p_2 + a_{31}p_3 + ... + a_{n1}p_n \geq v, \\ a_{12}p_1 + a_{22}p_2 + a_{32}p_3 + ... + a_{n2}p_n \geq v, \\ .... \\ a_{1m}p_1 + a_{2m}p_2 + a_{3m}p_3 + ... + a_{nm}p_n \geq v. \end{cases}$$

Помітимо, що $v > 0$, та, поділивши усі нерівності на v, перейдемо до величин $x_i = \dfrac{p_i}{v}$. Отримаємо систему нерівностей

$$\begin{cases} a_{11}x_1 + a_{21}x_2 + a_{31}x_3 + ... + a_{n1}x_n \geq 1, \\ a_{12}x_1 + a_{22}x_2 + a_{32}x_3 + ... + a_{n2}x_n \geq 1, \\ .... \\ a_{1m}x_1 + a_{2m}x_2 + a_{3m}x_3 + ... + a_{nm}x_n \geq 1. \end{cases}$$

Умова нормування $\sum_{i=1}^{n} p_i = 1$ для x_i набуває вигляду $\sum_{i=1}^{n} x_i = \frac{1}{v}$. Задача вибору оптимальної мішаної стратегії полягає у виборі таких чисел $(p_1, p_2, ..., p_n)$, щоб гарантований середній виграш v був би максимальним, тоді $\frac{1}{v}$ має бути мінімальним і задача набуває вигляду типової *задачі лінійного програмування*

$$\frac{1}{v} = x_1 + x_2 + x_3 + ... + x_n \to \min$$
$$\begin{cases} a_{11}x_1 + a_{21}x_2 + a_{31}x_3 + ... + a_{n1}x_n \geq 1, \\ a_{12}x_1 + a_{22}x_2 + a_{32}x_3 + ... + a_{n2}x_n \geq 1, \\ .... \\ a_{1m}x_1 + a_{2m}x_2 + a_{3m}x_3 + ... + a_{nm}x_n \geq 1, \end{cases} \quad (1.1)$$
$$x_i \geq 0, \quad i = 1...n.$$

Розв'язавши цю задачу (наприклад, симплекс-методом [7]), ми знайдемо v та $(x_1, x_2, ..., x_n)$, звідки потім знайдемо $p_i = x_i \cdot v$.

Аналогічними міркуваннями для знаходження мішаної стратегії другого гравця – ймовірностей ходів $(q_1, q_2, ..., q_m)$ та максимального середнього програшу w – отримуємо таку задачу:

$$\frac{1}{w} = y_1 + y_2 + y_3 + ... + y_m \to \max$$
$$\begin{cases} a_{11}y_1 + a_{12}y_2 + a_{13}y_3 + ... + a_{1m}y_m \leq 1, \\ a_{21}y_1 + a_{22}y_2 + a_{23}y_3 + ... + a_{2m}y_m \leq 1, \\ .... \\ a_{n1}y_1 + a_{n2}y_2 + a_{n3}y_3 + ... + a_{nm}y_m \leq 1, \end{cases} \quad (1.2)$$
$$y_j \geq 0, \quad j = 1...m,$$

розв'язавши яку знайдемо $q_j = y_j \cdot w$. Помітимо, що задачі (1.1) і (1.2) є парою двоїстих задач, тому за першою теоремою двоїстості екстремальні значення лінійних форм мають співпадати, тобто $v = w$. Це спільне значення v та w називається *ціною гри*. Системи (1.1) і (1.2) дозволяють знайти оптимальні мішані стратегії обох гравців та ціну гри, тобто *розв'язати гру*.

Основна теорема. У мішаних стратегіях кожна матрична гра двох осіб з нульовою сумою має сідлову точку.

Приклад. Розв'язати матричну гру

$$\begin{pmatrix} -1 & -2 & -1 & 0 & -2 \\ 1 & -2 & 0 & 3 & -1 \\ -2 & -2 & -1 & 1 & -2 \\ 2 & 0 & 1 & -1 & 0 \end{pmatrix}.$$

Розв'язання. У першу чергу спробуємо знайти розв'язок гри у чистих стратегіях. Для цього знайдемо та порівняємо мінімакс і максимін.

$$\begin{array}{c|c} \left(\begin{array}{ccccc} -1 & \underline{-2} & -1 & 0 & -2 \\ 1 & \underline{-2} & 0 & \overline{3} & -1 \\ -2 & \underline{-2} & -1 & 1 & -2 \\ \overline{2} & \overline{0} & \overline{1} & \underline{-1} & \overline{0} \end{array}\right) & \begin{array}{c} \to \min \\ -2 \\ -2 \\ -2 \\ \overline{-1} \end{array} \\ \hline \begin{array}{ccccc} \downarrow \max & & & & \\ 2 & \underline{0} & 1 & 3 & \underline{0} \end{array} & \begin{array}{c} -1 \\ 0 \end{array} \end{array}$$

Виписуємо з кожного рядка мінімальний елемент (підкреслений знизу), а з кожного стовпчика – максимальний (риска зверху). В отриманому стовпчику знаходимо максимум – це буде максимін, а в рядку – мінімум (тобто мінімакс). Отримали такі значення $\max\limits_i \min\limits_j a_{ij} = -1$, $\min\limits_j \max\limits_i a_{ij} = 0$, тобто $\max\limits_i \min\limits_j a_{ij} < \min\limits_j \max\limits_i a_{ij}$, а отже, розв'язку гри у чистих стратегіях не існує.

Перш ніж знаходити оптимальні мішані стратегії гравців, виконаємо *спрощення платіжної матриці*. Помітимо, що **якщо всі елементи деякого рядка матриці не перевершують відповідні елементи іншого рядка**, то стратегія з «меншим» рядком є невигідною першому гравцю, бо за будь-якого ходу другого гравця перший отримає менший виграш. Таким чином, стратегію, що відповідає меншому рядку, гравець не використовуватиме і **цей рядок з платіжної матриці можна «викреслити»** (ймовірність вибору цього рядка в остаточній відповіді буде дорівнювати 0). Так само **можна «викреслити» стовпчик, якщо його елементи не перебільшують відповідні елементи іншого стовпчика**, а у відповіді записати нульову ймовірність вибору відповідної стратегії. Спробуємо спростити таким чином нашу матрицю.

$$\begin{pmatrix} \cancel{-1} & \cancel{-2} & \cancel{-1} & \cancel{0} & \cancel{-2} \\ \cancel{1} & -2 & \cancel{0} & 3 & \cancel{-1} \\ \cancel{-2} & \cancel{-2} & \cancel{-1} & \cancel{1} & \cancel{-2} \\ \cancel{2} & 0 & \cancel{1} & -1 & \cancel{0} \end{pmatrix}$$

Порівняємо перший та другий рядки і помітимо, що елементи першого рядка менші за елементи другого, тому перший рядок викреслюємо. Також після порівняння другого і третього рядків викреслюємо третій рядок. Порівняння другого і четвертого рядків не дозволяє щось викреслити, бо $1<2$, $-2<0$, $0<1$, але $3>-1$. Тепер подивимося на стовпчики і викреслимо перший, третій та п’ятий, які більші за другий. Другий і четвертий стовпчики залишаються, бо $-2<3$, але $0>-1$. Таким чином, ми маємо розв’язати матричну гру з платіжною матрицею $\begin{pmatrix} -2 & 3 \\ 0 & -1 \end{pmatrix}$. Для зведення цієї задачі до ЗЛП треба додати до усіх елементів матриці однакове число, наприклад 3, що приведе нас до матриці з додатними елементами $\begin{pmatrix} 1 & 6 \\ 3 & 2 \end{pmatrix}$.

Позначимо через (p_1, p_2) мішану стратегію першого гравця, (q_1, q_2) – мішану стратегію другого гравця, v – ціну гри. Із системи (1.1) для невідомих $x_i = \dfrac{p_i}{v}$, $y_j = \dfrac{q_j}{w}$ запишемо пару двоїстих задач

$$\frac{1}{v} = x_1 + x_2 \to \min \qquad\qquad \frac{1}{w} = y_1 + y_2 \to \max$$

$$\begin{cases} x_1 + 3x_2 \geq 1, \\ 6x_1 + 2x_2 \geq 1, \end{cases} \qquad\qquad \begin{cases} y_1 + 6y_2 \leq 1, \\ 3y_1 + 2y_2 \leq 1, \end{cases}$$

$$x_{1,2} \geq 0, \qquad\qquad y_{1,2} \geq 0.$$

Розв'яжемо їх двоїстим симплекс-методом [7]. Складемо симплекс-таблицю для другої задачі:

	$\begin{matrix} x_3 \\ \updownarrow \\ -y_1 \end{matrix}$	$\begin{matrix} x_4 \\ \updownarrow \\ -y_2 \end{matrix}$	1
$x_1 \leftrightarrow y_3$	1	6	1
$x_2 \leftrightarrow y_4$	3	2	1
$\frac{1}{v} = \frac{1}{w}$	-1	-1	0

Зробивши перетворення таблиці за алгоритмом симплекс-методу, отримуємо таку симплекс-таблицю:

	$\begin{matrix} x_2 \\ \updownarrow \\ -y_4 \end{matrix}$	$\begin{matrix} x_1 \\ \updownarrow \\ -y_3 \end{matrix}$	1
$x_4 \leftrightarrow y_2$	*	*	$\frac{1}{8}$
$x_3 \leftrightarrow y_1$	*	*	$\frac{1}{4}$
$\frac{1}{v} = \frac{1}{w}$	$\frac{5}{16}$	$\frac{1}{16}$	$\frac{3}{8}$

З цієї таблиці знаходимо оптимальні розв'язки двоїстих задач $y_1 = \frac{1}{4}$, $y_2 = \frac{1}{8}$, $x_1 = \frac{1}{16}$, $x_2 = \frac{5}{16}$, $\frac{1}{v} = \frac{1}{w} = \frac{3}{8}$.

Таким чином, $v = w = \frac{8}{3}$, $q_1 = \frac{1}{4} \cdot \frac{8}{3} = \frac{2}{3}$, $q_2 = \frac{1}{8} \cdot \frac{8}{3} = \frac{1}{3}$,

$$p_1 = \frac{1}{16} \cdot \frac{8}{3} = \frac{1}{6}, \quad p_2 = \frac{5}{16} \cdot \frac{8}{3} = \frac{5}{6}.$$

Згадаємо тепер, що до елементів платіжної матриці ми додавали 3, збільшивши тим самим на 3 ціну гри, тобто ціна вихідної гри $\tilde{v} = \frac{8}{3} - 3 = -\frac{1}{3}$. Оптимальні мішані стратегії запишемо з урахуванням викреслених рядків та стовпчиків: для першого гравця $\left(0; \frac{1}{6}; 0; \frac{5}{6}\right)$ та для другого гравця $\left(0; \frac{2}{3}; 0; \frac{1}{3}; 0\right)$.

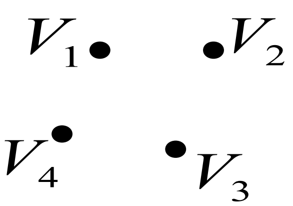
2 ПРОГРАМУВАННЯ НА СІТКАХ

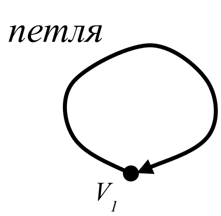
Деякі відомості про графи

Багато економічних задач, наприклад задачі про перевезення вантажу, перекачування нафти й газу по трубопроводах, управління запасами тощо, зручно формулювати у термінах сіток і потоків. Основою такого типу завдань є орієнтовані та неорієнтовані графи.

Граф – це сукупність множини точок $V_1, V_2, ..., V_n$, які називаються *вершинами* графа, та відрізків $u_1, ..., u_m$, які називаються *ребрами* графа, на площині або у просторі.

На рисунках 2.1-2.6 наведені деякі види графів.





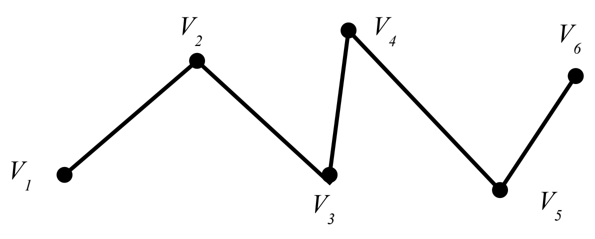


Рисунок 2.1 Рисунок 2.2 Рисунок 2.3

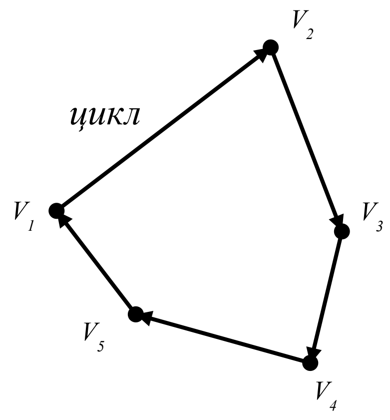


Рисунок 2.4

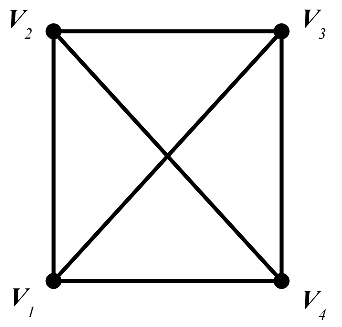


Рисунок 2.5

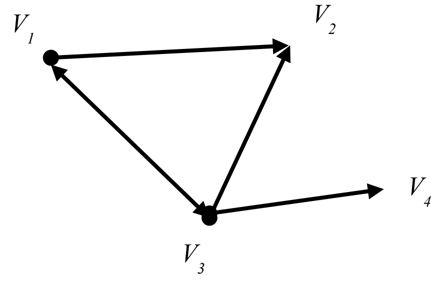


Рисунок 2.6

Дві вершини називаються *суміжними*, якщо існує ребро, що з'єднує ці вершини, наприклад, на рисунку 2.1 всі вершини несуміжні, на рисунку 2.3 вершини V_1 та V_2 суміжні, а вершини V_1 та V_4 – ні.

Граф називається *орієнтованим*, якщо вказаний напрямок ребер (рисунки 2.2, 2.4, 2.6) і *неорієнтованим*, якщо такий напрямок не вказаний (рисунки 2.1, 2.3, 2.5).

Граф називається *зв'язним*, якщо будь-які його дві вершини можна з'єднати послідовністю ребер (дивись рисунки 2.2–2.5).

З кожним ребром графа у задачах пов'язують деякі числові характеристики, наприклад, пропускну спроможність і т. ін. У таких випадках кажуть, що ребрам графа приписана певна вага, граф називають *зваженим*.

Потоки на сітках

Сітка – це зважений граф без жодної петлі та циклу, орієнтований в одному загальному напрямку від вершини V_1, яка розглядається як *вхід* (*джерело*) графа, до вершини V_n, яка є *виходом* (*стоком*) графа.

Припустимо, що по ребру можна пропустити деяку кількість речовини (вантаж, ресурс, інформація і т. ін.)

Максимальна кількість речовини, яку може пропустити за одиницю часу ребро $u_k = \{V_i, V_j\}$, називається його *пропускною спроможністю* і позначається r_{ij}. Пропускна спроможність у зворотному напрямку позначається r_{ji} (дивись рисунок 2.7).

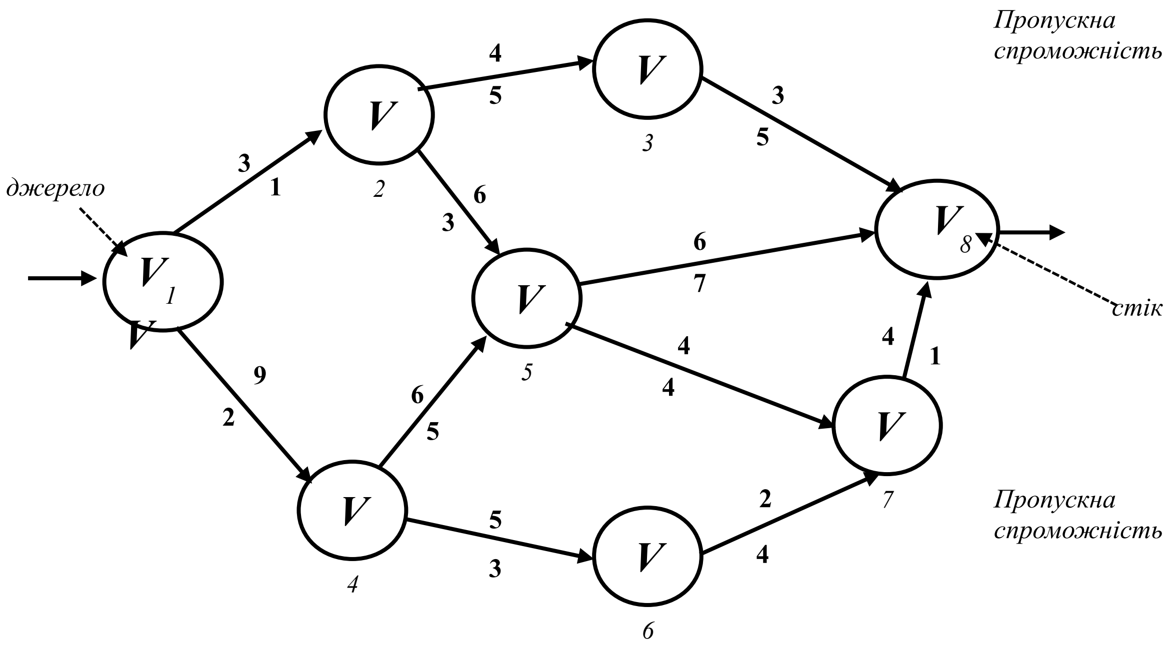


Рисунок 2.7

У загальному випадку $r_{ij} \neq r_{ji}$. Якщо вершини V_l та V_k не з'єднані ребром, то будемо вважати $r_{lk} = r_{kl} = 0$. Наприклад, для сітки на рисунку 2.7 пропускна спроможність $r_{15} = r_{51} = 0$.

Кількість речовини φ_{ij}, яка протікає по ребру в одиницю часу, називається *потоком по ребру*. Потік з вершини V_i у вершину V_j позначимо φ_{ij}.

Теорема. Потік по ребру не перевищує його пропускну спроможність

$$0 \le \varphi_{ij} \le r_{ij}. \tag{2.1}$$

Ребро називається *насиченим*, якщо потік по ньому дорівнює його пропускній спроможності, і *ненасиченим* в протилежному випадку.

Для будь-якої вершини, крім джерела і стоку, кількість речовини, яка надходить у цю вершину, дорівнює кількості речовини, яка витікає з неї (*умова збереження потоку*) – у проміжних вершинах потоки не створюються і не зникають

$$\sum_i \varphi_{ik} - \sum_j \varphi_{kj} = 0,\ \ k \neq 1, k \neq n. \qquad (2.2)$$

Звідси випливає, що загальна кількість речовини, яка витікає із джерела V_1, збігається із загальною кількістю речовини, яка надходить у стік V_n, тобто

$$f = \sum_j \varphi_{1j} = \sum_i \varphi_{in}. \qquad (2.3)$$

Лінійну функцію f називають *величиною потоку на сітці.*

Поставимо задачу *про відшукання максимального потоку на сітці*, тобто такого розв'язку $\left\{ \varphi_{ij}^{*} \right\}$ системи обмежень (2.1) – (2.2), який максимізує лінійну функцію (2.3).

Ця задача є *задачею лінійного програмування* та може бути розв'язана симплекс-методом. Окремим випадком цієї задачі є задача відшукання максимального потоку транспорту по заданій сітці. Оскільки в цьому випадку пропускні спроможності ребер є цілими числами, то виникає задача *цілочислового програмування.*

Зважаючи на специфічність обмежень задачі про максимальний потік, для її розв'язання були створені спеціальні методи, більш ефективні, ніж алгоритм симплекс-методу. Один з них ми розглянемо далі.

Розріз на сітці

Нехай задано деяку сітку. Розіб'ємо множину вершин сітки на дві множини A і B, які не перетинаються, так, щоб джерело потрапило в одну множину, а стік – в іншу. У цьому випадку говорять, що на сітці зроблений розріз, який відокремлює джерело від стоку (рисунок 2.8).

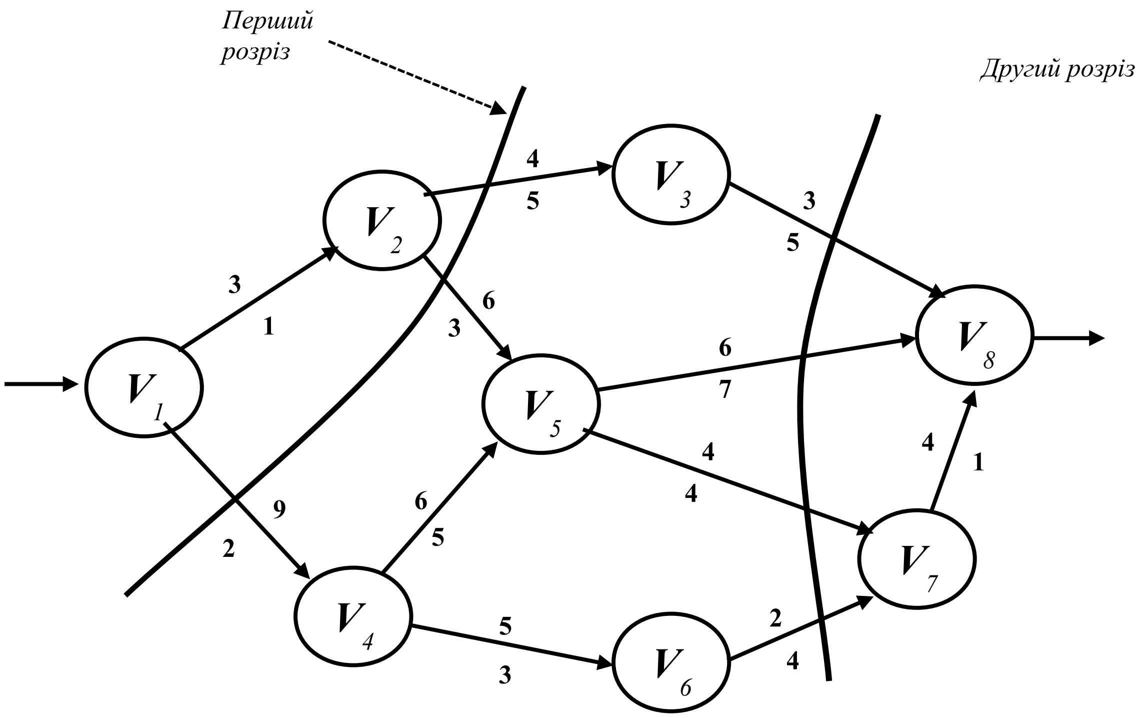


Рисунок 2.8

Сукупність ребер, початкові вершини яких потрапили у множину А, а кінцеві – в множину В, називається *розрізом.* Величина $R = \sum_{i \in A} \sum_{j \in B} r_{ij}$ називається *пропускною спроможністю розрізу*.

У наданому прикладі (дивись рисунок 2.8) пропускна спроможність першого розрізу $R = 4 + 6 + 9 = 19$, пропускна спроможність другого розрізу $R = 3 + 6 + 4 + 2 = 15$. Бачимо, що другий розріз має меншу пропускну спроможність.

Розріз з найменшою пропускною спроможністю назвемо *мінімальним розрізом*.

Теорема Форда-Фалкерсона. *На будь-якій сітці максимальна величина потоку з джерела у стік дорівнює пропускній спроможності мінімального розрізу*.

Задача відшукання мінімального розрізу є двоїстою до задачі про відшукання максимального потоку. Очевидно для її розв’язання треба перебрати всі можливі шляхи з джерела у стік, використовуючи повністю їх пропускні спроможності. При цьому, якщо ребро вже входило у деякий шлях, то воно може бути враховано і у наступних шляхах, але вже зі зміненою пропускною спроможністю.

Алгоритм відшукання максимального потоку (або мінімального розрізу) розглянемо на прикладі.

Приклад. Знайти максимальну пропускну спроможність сітки (рисунок 2.9).

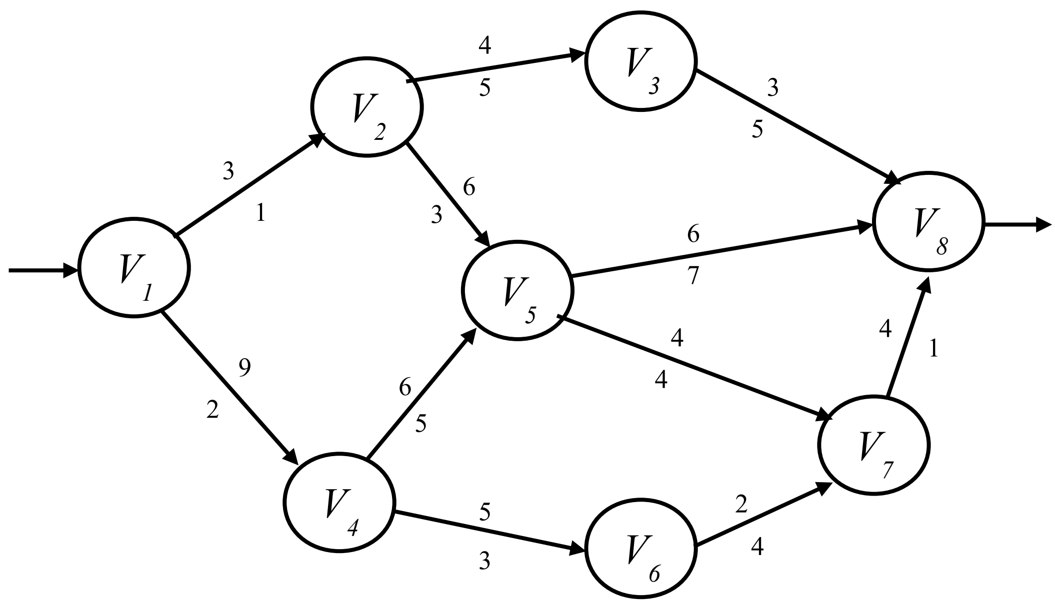
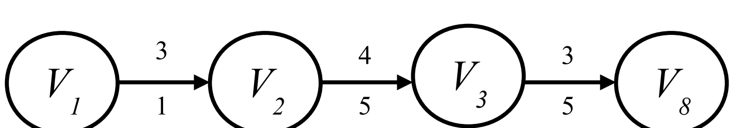


Рисунок 2.9

Розв’язання. Оберемо довільний шлях з джерела в стік, наприклад:



Обчислимо пропускну спроможність даного шляху $\theta_1 = \min\{3,4,3\} = 3$. Віднімаємо $\theta_1 = 3$ від відповідних r_{ij} над обраними ребрами, а до r_{ji} під обраними ребрами додаємо (рисунок 2.10).

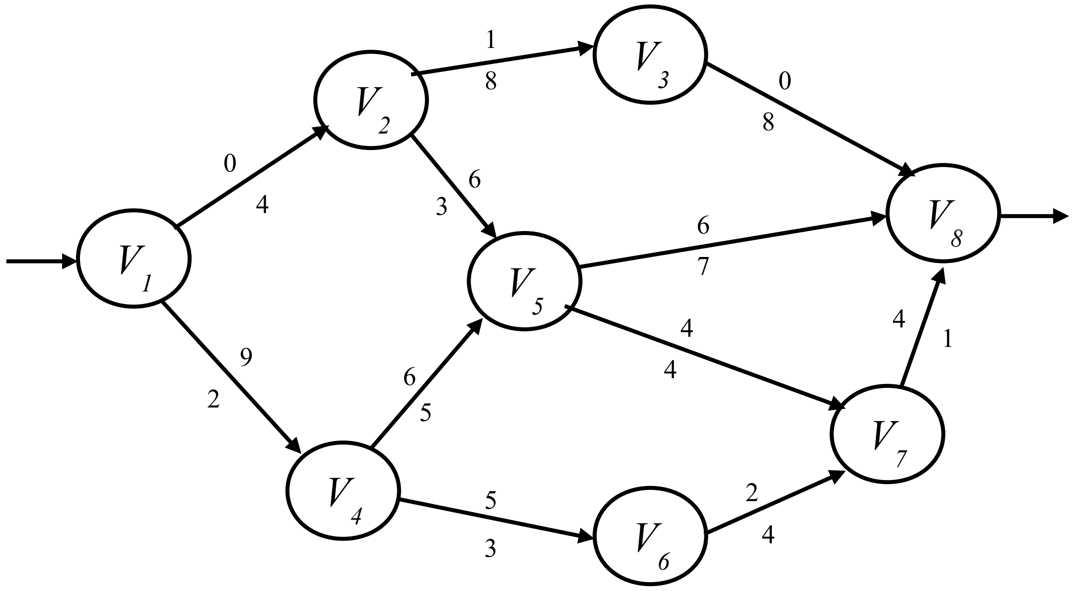
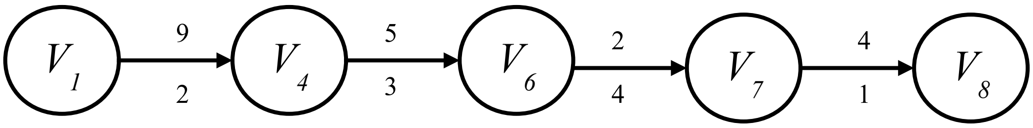


Рисунок 2.10

Ще існують шляхи з джерела у стік. Обираємо такий шлях:



Обчислюємо його пропускну спроможність $\theta_2 = \min\{9,5,2,4\} = 2$. Для кожного ребра шляху віднімаємо θ_2 від верхніх r_{ij} та додаємо до нижніх r_{ji} (рисунок 2.11).

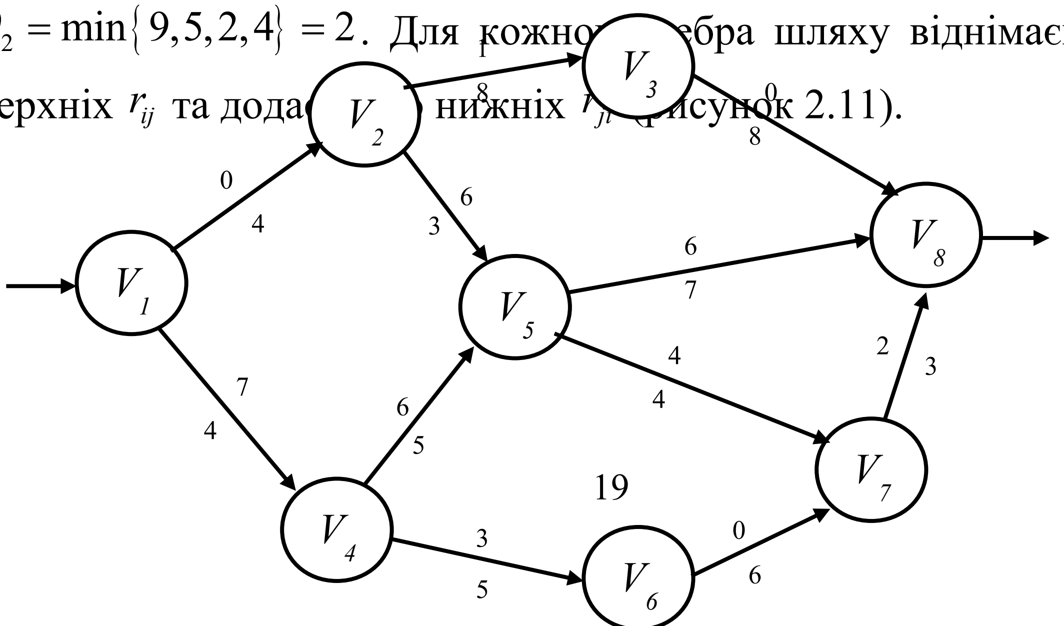
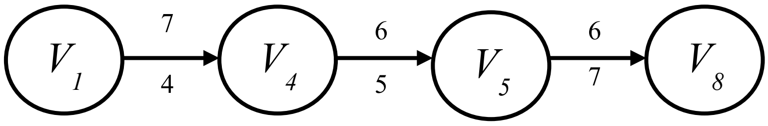


Рисунок 2.11

Ще один можливий шлях виглядає так:



Його пропускна спроможність $\theta_3 = \min\{7,6,6\} = 6$. Рахуємо нову пропускну спроможність кожного обраного ребра (рисунок 2.12).

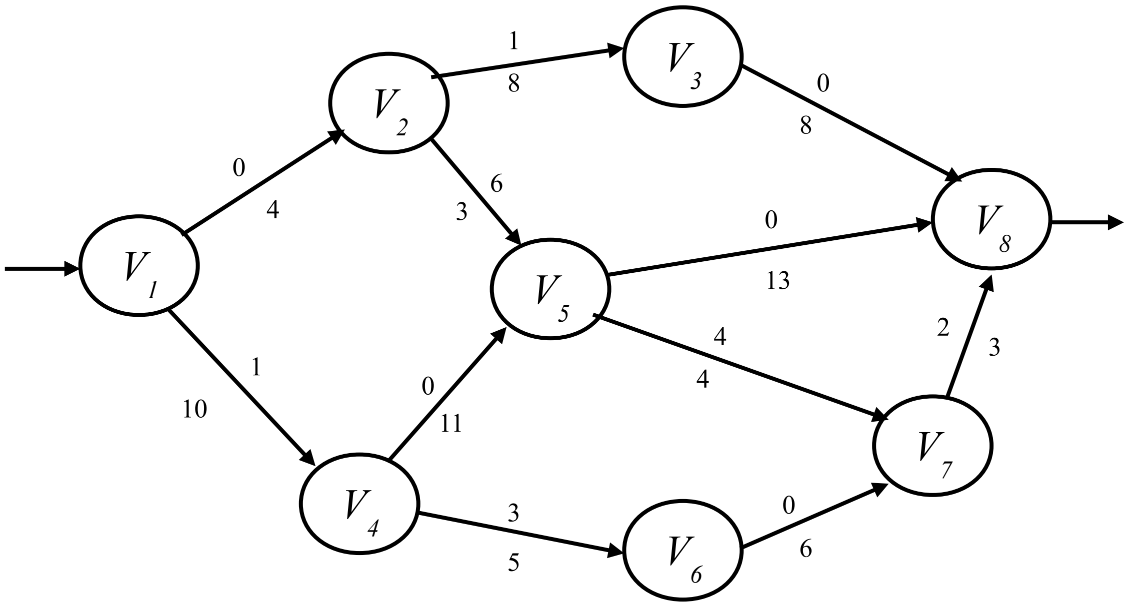


Рисунок 2.12

Більше немає шляхів із джерела в стік. Нарисуємо *мінімальний розріз*, який *відокремлює джерело від стоку* та *проходить через ребра з нульовою пропускною спроможністю* (рисунок 2.13).

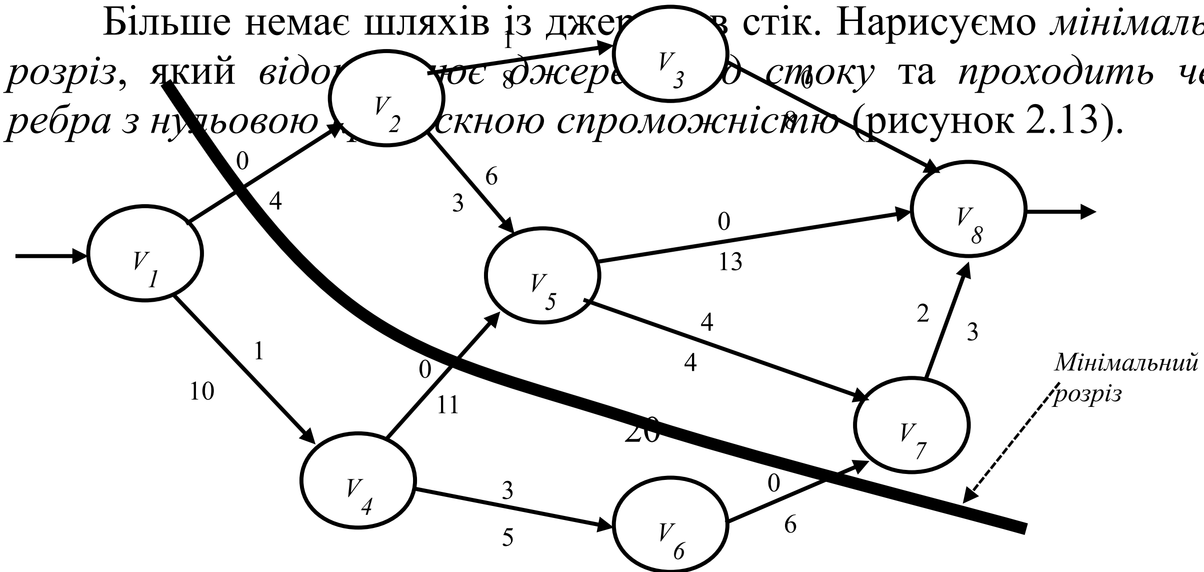


Рисунок 2.13

Для визначення отриманого *потоку* віднімемо від пропускних спроможностей першої сітки (рисунок 2.9) пропускні спроможності останньої сітки (рисунок 2.13), замінюючи від'ємні числа нулями (рисунок 2.14).

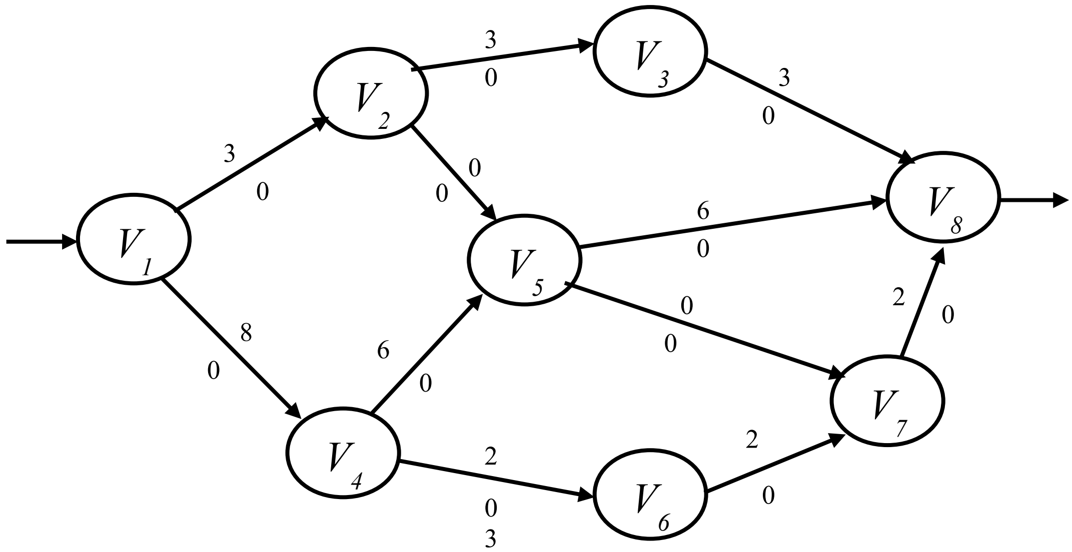


Рисунок 2.14

За формулою (2.3) обчислюємо максимальний потік по сітці $f = \varphi_{14} + \varphi_{12} = 8 + 3 = 11$ або $f = \varphi_{38} + \varphi_{58} + \varphi_{78} = 3 + 6 + 2 = 11$.

З іншого боку, перенесемо мінімальний розріз (дивись рисунок 2.13) на першу сітку (дивись рисунок 2.9).

Зауваження. Порівняємо рисунки 2.14 та 2.15. Бачимо, що мінімальний розріз проходить по насичених ребрах.

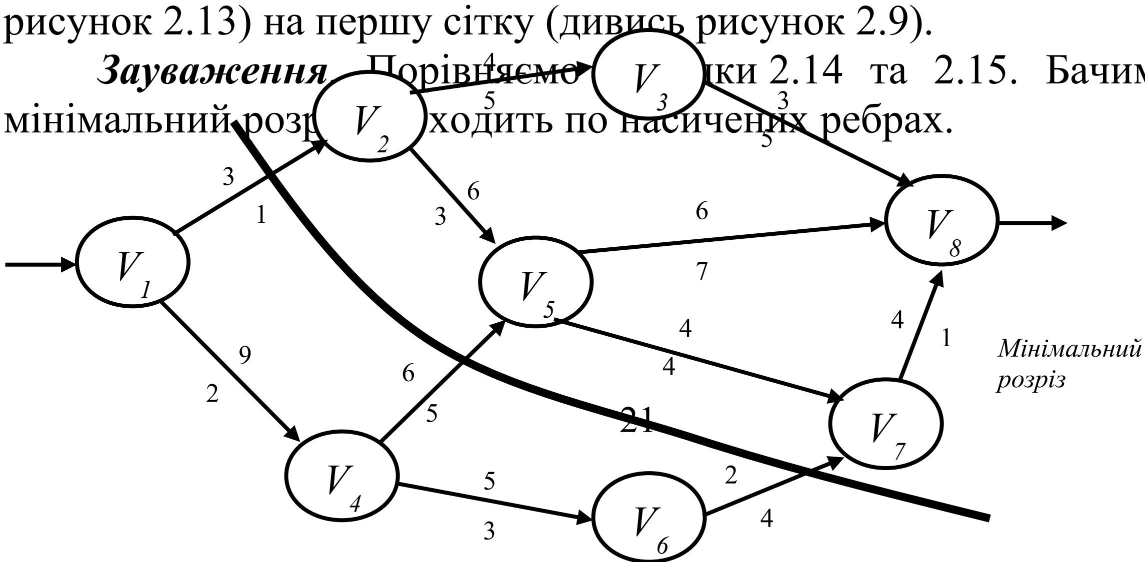


Рисунок 15

За теоремою Форда-Фалкерсона максимальна пропускна спроможність сітки дорівнює пропускній спроможності мінімального розрізу $R = 3 + 6 + 2 = 11$. Як бачимо, результат співпадає із знайденим раніше.

3 НЕЛІНІЙНЕ ПРОГРАМУВАННЯ

Градієнт функції

Градієнтом ($gradf, \overline{\nabla f}$) функції $z = f(x, y)$ в точці $M_0(x_0, y_0)$ називається вектор, координатами якого є частинні похідні функції $z = f(x, y)$ в точці $M_0(x_0, y_0)$, тобто

$gradf\,|_{M_0} = f'_x(M_0)\vec{i} + f'_y(M_0)\vec{j}$ або $gradf\,|_{M_0} = \left(f'_x(M_0), f'_y(M_0)\right)$.

Довжина (норма, модуль) градієнта дорівнює $|gradf| = \sqrt{(f'_x)^2 + (f'_y)^2}$.

Напрям градієнта є напрямом найшвидшого зростання функції $z = f(x, y)$, модуль градієнта дорівнює найбільшій швидкості зростання функції $z = f(x, y)$.

Градієнт функції

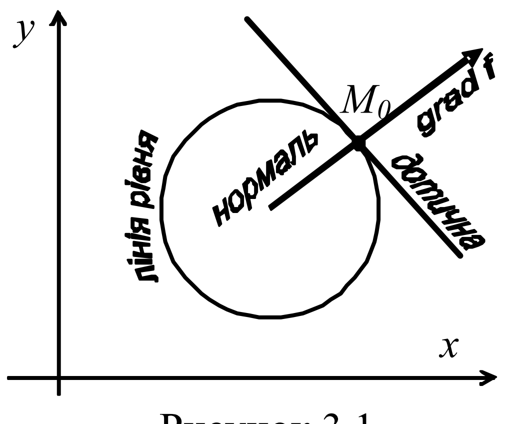


Рисунок 3.1

$z = f(x, y)$ в кожній точці $M_0(x_0, y_0)$ розташований вздовж нормалі до лінії рівня поверхні $z = f(x, y)$, яка проходить через точку $M_0(x_0, y_0)$ (рисунок 3.1). (Лінією рівня називається лінія, на якій значення функції $z = f(x, y)$ виявляються однаковими сталими величинами, тобто лінія рівня визначається рівнянням $f(x, y) = const$).

Якщо функція $f(\overline{X}) = f(x_1, x_2, ..., x_n)$ диференційована в точці $\overline{X}^0 = \left(x_1^0, x_2^0, ..., x_n^0\right)$, тоді *градієнтом* функції $f(\overline{X})$ в точці $\overline{X}^0$ називається *n*-вимірний вектор з координатами

$$gradf\,\Big|_{\overline{X}^0} = \left(f'_{x_1}\left(\overline{X}^0\right), f'_{x_2}\left(\overline{X}^0\right), ..., f'_{x_n}\left(\overline{X}^0\right)\right).$$

Всі властивості градієнта зберігаються і в *n*-вимірному просторі.

Вектор ($-gradf$) називають *антиградієнтом*. Він вказує напрям найшвидшого спадання функції.

Опуклі множини та функції

Множина точок *G* називається *опуклою*, якщо разом з будь-якими двома точками $M_1 = \left(x_1^1, x_2^1, ..., x_n^1\right) \in G$ та $M_2 = \left(x_1^2, x_2^2, ..., x_n^2\right) \in G$ їй належать усі точки відрізка $M_1 M_2$.

Тобто з умови $M_1 = \left(x_1^1, x_2^1, ... x_n^1\right) \in G$ та $M_2 = \left(x_1^2, x_2^2, ... x_n^2\right) \in G$ випливає, що точка $M = \lambda M_1 + (1 - \lambda) M_2$ ($\lambda \in [0,1]$) теж належить множині *G*. Прикладами опуклих множин на площині є відрізок, пряма, трикутник, круг та ін.

Функція $f(\overline{X})$, визначена на опуклій множині *G*, називається *опуклою* (рисунок 3.2), якщо для будь-яких точок $X_1 \in G$ та $X_2 \in G$ і $\lambda \in [0,1]$ справедлива нерівність

$$f\left(\lambda X_1 + (1-\lambda)X_2\right) \le \lambda f\left(X_1\right) + (1-\lambda) f\left(X_2\right). \qquad (3.1)$$

Якщо

$$f\left(\lambda X_1 + (1-\lambda)X_2\right) \ge \lambda f\left(X_1\right) + (1-\lambda) f\left(X_2\right), \qquad (3.2)$$

то функція $f\left(\overline{X}\right)$ називається *угнутою* (рисунок 3.3).

Якщо у співвідношеннях (3.1), (3.2) для $\lambda \in (0,1)$ наявні строгі нерівності, тоді функція називається *строго опуклою* (*строго угнутою*). Геометрична інтерпретація для двовимірного простору подана на рисунках 3.2, 3.3.

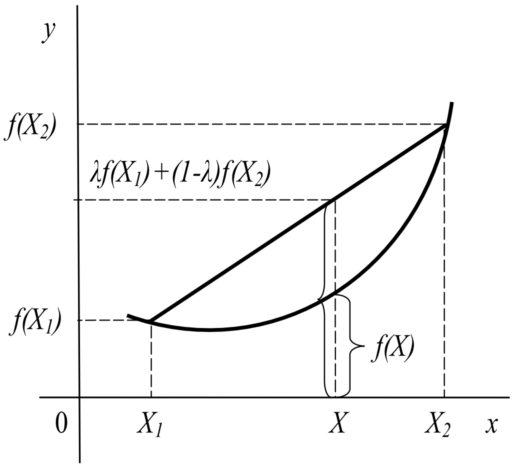


Рисунок 3.2

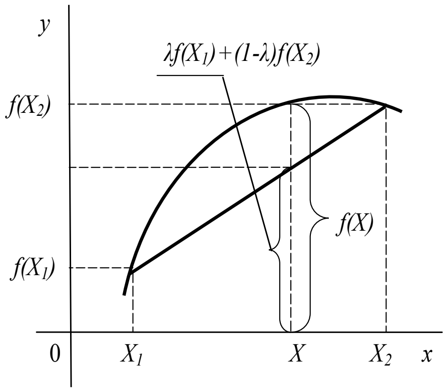


Рисунок 3.3

Невід’ємна лінійна комбінація опуклих на деякій опуклій множині G функцій, тобто функція $f = \sum_{i=1}^{n} \lambda_i f_i(x),\ \lambda_i \ge 0$ теж буде опуклою.

Властивості опуклих функцій

1 Якщо $\varphi(\overline{X})$ - опукла функція при усіх $\overline{X}(x_j \geq 0)$, тоді опуклою буде й множина розв'язків системи $\varphi(\overline{X}) \leq b$.

Аналогічна властивість існує і для угнутих функцій.

Зауваження. Множина розв'язків системи нерівностей

$$\varphi_i(\overline{X}) \leq b_i, \,^{-}(x_j \geq 0), i = \overline{1,m}, \; j = \overline{1,n} \tag{3.3}$$

або

$$\varphi_i(\overline{X}) \geq b_i, (x_j \geq 0), i = \overline{1,m}, \; j = \overline{1,n} \tag{3.4}$$

опукла, якщо в нерівностях (3.3) $\varphi_i(\overline{X})$ - опуклі функції, а в нерівностях (4.4) $\varphi_i(\overline{X})$ - угнуті функції.

2 Опукла функція $f(\overline{X})$, визначена на опуклій множині *G*, досягає свого локального мінімуму в кожній точці $\overline{X}$, в якій градієнт дорівнює нулю.

3 Локальний мінімум опуклої функції $f(\overline{X})$, яка визначена на опуклій множині *G*, збігається з її глобальним мінімумом на цій множині.

Задача нелінійного програмування

Задача математичного програмування

$$\begin{cases} \max(\min) z = f(\overline{X}) & (4.5) \\ \varphi_i(\overline{X})\{\leq,=,\geq\} b_i, \; i = \overline{1,m}, & (4.6) \\ \overline{X} = (x_1, x_2, ..., x_n), x_j \geq 0, \; j = \overline{1,n} \end{cases}$$

в якій цільова функція (3.5), або обмеження (3.6), або те й інше нелінійні, називається *нелінійною*.

Задачі нелінійного програмування з двома змінними можна, як і у випадку лінійного програмування, розв'язувати графічно, а саме: вважаючи $f(x_1, x_2) = C$, ми отримуємо рівняння сім'ї ліній рівня функції *f*; кожному фіксованому значенню параметра *C* відповідає певна лінія сім'ї.

Тоді задача про відшукання в області D (область припустимих розв'язків) такої точки, в якій функція f набуває найменшого (найбільшого) значення, зводиться до відшукання серед ліній рівня функції f, які перерізають область D, такої лінії, для якої параметр C має найменше (найбільше) значення. Точки перетину цієї лінії з областю D і будуть шуканими: в них будемо мати $f_{\min} = C_{\min} \left(f_{\max} = C_{\max} \right)$. Розглянемо цю задачу на прикладі.

Приклад. В області D, яка визначається системою нерівностей

$$\begin{cases} 3x_1 + x_2 \le 29 \\ x_1 - x_2 \le 3 \\ 5x_1 - x_2 \ge 19, \;\; x_1 \ge 0, x_2 \ge 0 \end{cases}$$

визначити точку $X^0\left(x_1^0, x_2^0\right)$, в якій функція $f\left(x_1, x_2\right)$ набуває:

а) найменшого значення; б) найбільшого значення, якщо:

1) $f\left(x_1, x_2\right) = \left(x_1 - 8\right)^2 + \left(x_2 - 1\right)^2$;

2) $f\left(x_1, x_2\right) = \left(x_1 - 7\right)^2 + \left(x_2 - 6\right)^2$.

Розв'язання. Побудуємо область D (рисунок 3.4). Це буде множина точок, які належать трикутнику ABC [7].

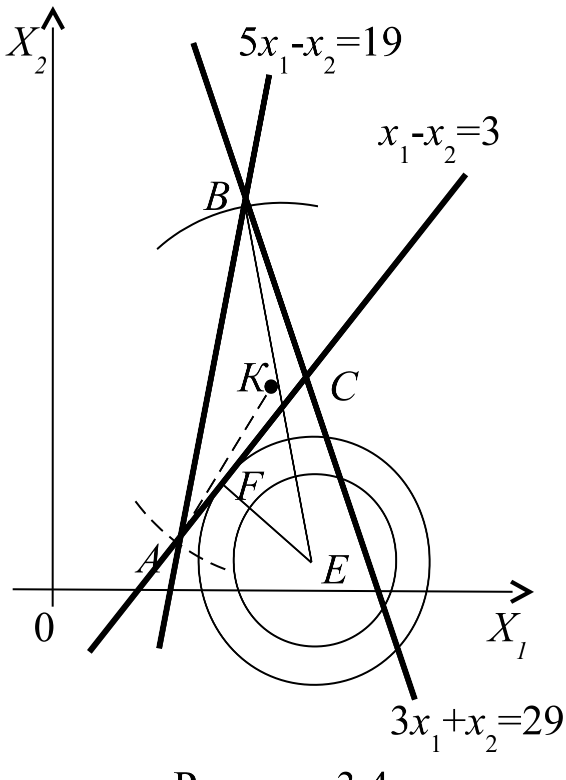


Рисунок 3.4

1а) $f = \left(x_1 - 8\right)^2 + \left(x_2 - 1\right)^2 \to \min$

Сім'єю ліній рівня функції f буде сім'я концентричних кіл

$$\left(x_1 - 8\right)^2 + \left(x_2 - 1\right)^2 = C$$

із загальним центром в точці $E(8,1)$.

Лінія рівня, для якої $f_{\min} = C_{\min}$ буде дотикатися в точці F границі

AC області D.

Обчислимо координати точки F дотику, для чого складемо рівняння прямої EF, перпендикулярної до AC.

(Якщо пряма (L) має загальне рівняння $Ax + By + C = 0$, тоді вектор $\overline{N}(A,B) \perp L$; отже, вектор $\overline{N}$ паралельний перпендикуляру до L і рівняння перпендикуляру має вигляд $\dfrac{x - x_0}{A} = \dfrac{y - y_0}{B}$, де точка $M_0(x_0, y_0)$ належить цьому перпендикуляру).

В нашому випадку, рівняння EF має вигляд

$$\frac{x-8}{1} = \frac{y-1}{-1} \Rightarrow -x + 8 = y - 1 \Rightarrow x + y - 9 = 0 \text{ або } x_1 + x_2 - 9 = 0.$$

Далі обчислимо координати точки F, як точки перетину прямих AC та EF

$$\begin{matrix}(AC)\\(EF)\end{matrix}\begin{cases}x_1 - x_2 = 3\\ x_1 + x_2 = 9\end{cases} \Rightarrow x_1 = 6, x_2 = 3$$

Таким чином, розв’язок задачі 1а такий:

$$X^0(6,3),\ f_{\min} = f(6,3) = (6-8)^2 + (3-1)^2 = 4 + 4 = 8.$$

1б) $f = (x_1 - 8)^2 + (x_2 - 1)^2 \to \max$

Точка перетину області D з лінією рівня функції f, яка відповідає $f_{\max}$, міститься у вершині B області D і має координати

$$\begin{matrix}(AB)\\(BC)\end{matrix}\begin{cases}5x_1 - x_2 = 19\\ 3x_1 + x_2 = 29\end{cases} \Rightarrow x_1 = 6, x_2 = 11$$

Таким чином, розв’язок задачі (б) має вигляд:

$$X^0(6{,}11)\ ,\ f_{\max} = f(6{,}11) = (6-8)^2 + (11-1)^2 = 4 + 100 = 104.$$

2а) $f = (x_1 - 7)^2 + (x_2 - 6)^2 \to \min$

У цьому випадку центр ліній рівня $(x_1 - 7)^2 + (x_2 - 6)^2 = C$ функції $f(x_1, x_2) = (x_1 - 7)^2 + (x_2 - 6)^2$ розташований всередині області *D*- це точка *К*(7,6). Очевидно, що найменше значення функції f, яке дорівнює нулю, вона має в точці *К*, тобто

$$X^0(7{,}6),\ f_{\min} = f(7{,}6) = (7-7)^2 + (6-6)^2 = 0.$$

2б) $f = (x_1 - 7)^2 + (x_2 - 6)^2 \to \max$

Найбільшого значення функція *f* досягає в точці *A* на межі області *D*. Знайдемо координати точки *A*:

$$\begin{matrix}(AB)\\(AC)\end{matrix}\begin{cases}5x_1 - x_2 = 19\\ x_1 - x_2 = 3\end{cases} \Rightarrow x_1 = 4, x_2 = 1 \Rightarrow A(4{,}1)$$

Таким чином, розв’язок задачі 2б має вигляд:

$$X^0(4{,}1)\ ,\ f_{\min} = f(4{,}1) = (4-7)^2 + (1-6)^2 = 34.$$

З цих прикладів випливає, що в нелінійному випадку екстремальною точкою може бути будь-яка точка області припустимих розв’язків (внутрішня, межова, кутова).

Метод множників Лагранжа

Розглянемо класичну задачу оптимізації

$$\begin{cases} \max(\min) z = f(\overline{X}) \\ \varphi_i(\overline{X}) = b_i,\ i = \overline{1,m}, \qquad \overline{X} = (x_1, x_2, ..., x_n), \quad m < n \end{cases} \qquad (4.7)$$

Ця задача відрізняється тим, що серед обмежень задачі (3.7) немає нерівностей, немає умови невід'ємності змінних і функції $f(x), \varphi_i(x)$ неперервні і мають частинні похідні принаймні другого порядку.

Класичний підхід до розв'язання задачі (3.7) полягає в застосуванні функції Лагранжа, а саме: розглядається функція

$$L(x_1, x_2, ..., x_n, \lambda_1, \lambda_2, ..., \lambda_m) = f(x_1, x_2, ..., x_n) + \sum_{i=1}^{m} \lambda_i (b_i - \varphi_i(x_1, x_2, ..., x_n)),$$

яка називається *функцією Лагранжа*, а величини $\lambda_1, \lambda_2, ..., \lambda_m$- *множниками Лагранжа*.

Теорема. Необхідна умова існування локального екстремуму, якщо функція $f(x_1, x_2, ..., x_n)$ в точці $(\overline{X}, \Lambda)$ має умовний екстремум, то

$$\begin{cases} \dfrac{\partial L(\overline{X}, \Lambda)}{\partial x_j} = 0,\ j = \overline{1,n} \\ \dfrac{\partial L(\overline{X}, \Lambda)}{\partial \lambda_i} = b_i - \varphi_i(x_1, x_2, ..., x_n) = 0,\ i = \overline{1,m} \end{cases} \qquad (4.8)$$

Економічний зміст множників Лагранжа

Розглянемо найпростіший випадок задачі:

$$\begin{cases} \max(\min) z = f(x_1, x_2) & (4.9) \\ \varphi(x_1, x_2) = b & (4.10) \end{cases}$$

Припустимо, що умовний екстремум досягається в точці $X^* = (x_1^*, x_2^*)$, тоді відповідне екстремальне значення функції

$f^* = f\left(x_1^*, x_2^*\right)$. Припустимо далі, що в обмеженнях (3.7) величина “b” може змінюватися, тоді координати $\left(x_1^*, x_2^*\right)$ точки екстремуму, отже, і екстремальне значення f^* функції $z = \varphi(x_1, x_2)$ буде залежати від “b”, тобто $x_1^* = x_1^*(b)$, $x_2^* = x_2^*(b)$, $f^* = f\left(x_1^*(b), x_2^*(b)\right)$.

Визначимо похідну $$\frac{df^*}{db} = \frac{\partial f}{\partial x_1^*}\frac{dx_1^*}{db} + \frac{\partial f}{\partial x_2^*}\frac{dx_2^*}{db}. \qquad (3.11)$$

В силу рівності (3.10): $\varphi\left(x_1^*(b), x_2^*(b)\right) = b$. Звідси після диференціювання маємо $$\frac{\partial \varphi}{\partial x_1^*}\frac{dx_1^*}{db} + \frac{\partial \varphi}{\partial x_2^*}\frac{dx_2^*}{db} = 1. \qquad (3.12)$$

Крім того, в точці екстремуму X^* виконуються необхідні умови (3.8). Для $n = 2, m = 1$ отримуємо

$$\frac{\partial f}{\partial x_1^*} = \lambda \frac{\partial \varphi}{\partial x_1^*}; \frac{\partial f}{\partial x_2^*} = \lambda \frac{\partial \varphi}{\partial x_2^*} \qquad (3.13)$$

Підставляючи вирази (3.13) в (3.11) та враховуючи співвідношення (3.12), знаходимо

$$\frac{df^*}{db} = \lambda \frac{\partial \varphi}{\partial x_1^*}\frac{dx_1^*}{db} + \lambda \frac{\partial \varphi}{\partial x_2^*}\frac{dx_2^*}{db} = \lambda\left(\frac{\partial \varphi}{\partial x_1^*}\frac{dx_1^*}{db} + \frac{\partial \varphi}{\partial x_2^*}\frac{dx_2^*}{db}\right) = \lambda \Rightarrow \frac{df^*}{db} = \lambda.$$

Аналогічно для задачі (3.7) отримуємо $\frac{df^*}{db_i} = \lambda_i$, $i = \overline{1, m}$.

Якщо розглядати “f” як прибуток або вартість, а b_i - як об’єми деяких ресурсів , тоді множники Лагранжа λ_i показують, як зміниться максимальний прибуток (або мінімальна вартість), якщо кількість ресурсу i-го типу зменшиться на одиницю.

Теорема Куна – Такера

Узагальненням методу множників Лагранжа є теорема Куна – Такера на випадок, коли в нелінійній задачі, крім обмежень – рівностей, містяться також і нерівності. (Теорема Куна – Такера використовується в *теорії опуклого програмування*).

Якщо в задачі математичного програмування цільова функція $f(x_1, x_2, ..., x_n)$ опукла (угнута) і область *D*, в якій потрібно знайти екстремальне значення цієї функції, теж опукла (угнута), тоді це буде *задачею опуклого програмування*. У цьому випадку локальний мінімум опуклої цільової функції (локальний максимум угнутої цільової функції) виявляється і глобальним.

Розглянемо *задачу опуклого програмування*

$$\begin{cases} z = f(\overline{X}) \to \max \\ \varphi_i(\overline{X}) \ge 0,\ i = \overline{1,m} \\ \overline{X} = (x_1, x_2, ..., x_n),\ x_j \ge 0,\ j = \overline{1,n} \end{cases} \qquad (3.14)$$

Для скорочення запису позначимо

$$\Lambda = (\lambda_1, \lambda_2, ..., \lambda_m);\ (\overline{X}, \Lambda) = (x_1, x_2, ..., x_n, \lambda_1, \lambda_2, ..., \lambda_m),$$

$$L(\overline{X}, \Lambda) = f(\overline{X}) + \sum_{i=1}^{m} \lambda_i \varphi_i(\overline{X}) \text{ - } \textit{функція Лагранжа.} \qquad (3.15)$$

Точка $\left(\overline{X}^0, \Lambda^0\right)$ називається *сідловою точкою* функції Лагранжа $L(\overline{X}, \Lambda)$, якщо для будь-яких $\overline{X}(x_j \ge 0)$ і $\Lambda(\lambda_j \ge 0)$ наявна умова

$$L\left(\overline{X}, \Lambda^0\right) \le L\left(\overline{X}^0, \Lambda^0\right) \le L\left(\overline{X}^0, \Lambda\right). \qquad (3.16)$$

Область *D* задовольняє *умови регулярності Слейтера*, якщо існує принаймні одна точка $\overline{X} \in D$ така, що $\varphi_i(\overline{X}) > 0,\ i = \overline{1,m}$.

Теорема Куна – Такера. Точка $\overline{X}^0$ є оптимальним розв'язком задачі (3.14), область *D* якої задовольняє *умови регулярності*, тоді і тільки тоді, коли існує така точка Λ^0, що пара $\left(\overline{X}^0, \Lambda^0\right)$ є сідловою точкою функції Лагранжа (3.15)

Зауваження. Якщо область *D* визначається лише лінійними нерівностями, тоді теорема Куна – Такера буде правильною і без умови регулярності області припустимих розв'язків.

Зауваження. Якщо функції $f\left(\overline{X}\right)$ та $\varphi_i\left(\overline{X}\right)$ задачі (3.14) диференційовані, тоді умови (3.16) можна замінити такими

$$\begin{cases} \dfrac{\partial L\left(\overline{X}^0, \Lambda^0\right)}{\partial x_j} \begin{cases} < 0, \text{ якщо } x_j^0 = 0; \\ = 0, \text{ якщо } x_j^0 > 0, \quad j = \overline{1, n}; \end{cases} \\ \dfrac{\partial L\left(\overline{X}^0, \Lambda^0\right)}{\partial \lambda_i} \begin{cases} > 0, \text{ якщо } \lambda_i^0 = 0; \\ = 0, \text{ якщо } \lambda_i^0 > 0, \quad i = \overline{1, m}, \end{cases} \end{cases} \tag{4.17}$$

які називаються *умовами Куна – Такера*.

Зауваження. Задача (3.14) сформульована як задача на відшукання максимуму функції *f*. У випадку задачі на відшукання мінімуму функції *g* для застосування наведених співвідношень треба перейти від мінімізації до максимізації $(f = -g \to \max)$.

В умовах, які визначають область припустимих розв'язків задачі (3.14), використовуються тільки нерівності типу $\varphi_i\left(\overline{X}\right) \geq 0$. Якщо серед умов є нерівності типу $\varphi_i\left(\overline{X}\right) \leq 0$, то потрібно їх попередньо переписати у вигляді $\left(-\varphi_i\left(\overline{X}\right)\right) \geq 0$.

Приклад. Дано

$$g = (x_1 - 5)^2 + (x_2 - 1)^2 \to \min$$

$$\begin{cases} x_1 - 2x_2 \leq -2 \\ 7x_1 - 2x_2 \geq 13 \\ x_1 + 3x_2 \leq 21, \quad x_1 \geq 0, \ x_2 \geq 0 \end{cases}$$

Потрібно:

1) знайти розв'язок задачі графічним способом;

2) написати функцію Лагранжа даної задачі і знайти її сідлову точку, використовуючи розв'язок задачі, отриманий графічно.

Розв'язання

1 Побудуємо область D (рисунок 3.5). Це буде множина точок, які належать трикутнику ABC. Сім'єю ліній рівня функції $g=(x_1-5)^2+(x_2-1)^2$ буде сім'я концентричних кіл $(x_1-5)^2+(x_2-1)^2=C$ із центром в точці $E(5,1)$.

Параметр $C=R^2$, де R- змінний радіус кола сім'ї.

Лінією рівня, яка визначає ming, буде коло, яке дотикається до сторони AC трикутника ABC. Розв'язком задачі є координати точки дотику F $\min g=C_{\min}=R_{\min}^2$, де $R_{\min}=EF$. Рівняння прямої EF має вигляд (див. попередній приклад 1а)

$$\frac{x_1-5}{1}=\frac{x_2-1}{-2}\Rightarrow -2x_1+10=x_2-1\Rightarrow\ 2x_1+x_2-11=0$$

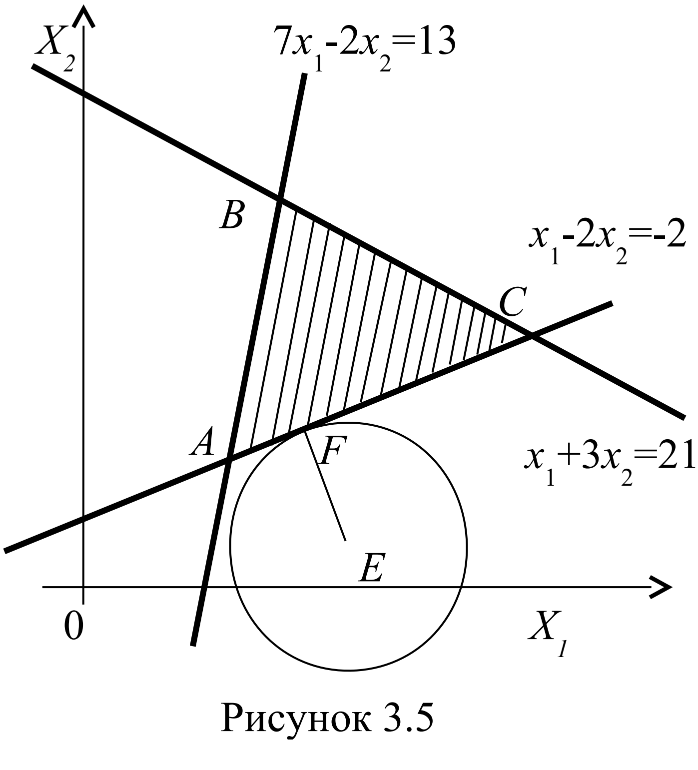


Рисунок 3.5

Координати точки перетину прямих AC та EF

$$\begin{matrix}(AC)\\(EF)\end{matrix}\begin{cases}x_1-2x_2=-2\\2x_1+x_2=11\end{cases}$$
$$\Rightarrow x_1=4, x_2=3$$

Таким чином, розв'язок задачі 2 такий:

$F(4,3)$,

$$g_{min}=C_{min}=g(4,3)=\\=(4-5)^2+(3-1)^2=5$$

2 Перепишемо задачу у вигляді

$$f = -g = -(x_1 - 5)^2 - (x_2 - 1)^2 \to \max$$
$$\begin{cases} -x_1 + 2x_2 - 2 \ge 0 \\ 7x_1 - 2x_2 - 13 \ge 0 \\ -x_1 - 3x_2 + 21 \ge 0, \quad x_1 \ge 0,\ x_2 \ge 0 \end{cases}$$

Розв'язок задачі $\overline{X}^0(4{,}3)$ при цьому залишається незмінним.

Функція Лагранжа має вигляд
$$L(\overline{X}, \Lambda) = -(x_1 - 5)^2 - (x_2 - 1)^2 + \lambda_1(-x_1 + 2x_2 - 2) + \lambda_2(7x_1 - 2x_2 - 13) +$$
$$+ \lambda_3(-x_1 - 3x_2 + 21).$$
її частинні похідні

$$\frac{\partial L}{\partial x_1} = -2(x_1 - 5) - \lambda_1 + 7\lambda_2 - \lambda_3, \quad \frac{\partial L}{\partial x_2} = -2(x_2 - 1) + 2\lambda_1 - 2\lambda_2 - 3\lambda_3$$
$$\frac{\partial L}{\partial \lambda_1} = -x_1 + 2x_2 - 2, \quad \frac{\partial L}{\partial \lambda_2} = 7x_1 - 2x_2 - 13, \quad \frac{\partial L}{\partial \lambda_2} = -x_1 - 3x_2 + 21.$$

Використовуємо далі умови (3.17), а також знайдену точку $\overline{X}^0(4{,}3)$ і отримуємо систему рівнянь відносно невідомих $\lambda_1, \lambda_2, \lambda_3$.

$$-2(4 - 5) - \lambda_1 + 7\lambda_2 - \lambda_3 = 0, \quad -2(3 - 1) + 2\lambda_1 - 2\lambda_2 - 3\lambda_3 = 0$$
$$\lambda_1(-4 + 2 \cdot 3 - 2) = 0,\ \lambda_2(7 \cdot 4 - 2 \cdot 3 - 13) = 0,\ \lambda_3(-4 - 3 \cdot 3 + 21) = 0,$$

Тобто систему

$$\begin{cases} 2 - \lambda_1 + 7\lambda_2 - \lambda_3 = 0, \\ -4 + 2\lambda_1 - 2\lambda_2 - 3\lambda_3 = 0 \\ 9\lambda_2 = 0, \quad 8\lambda_3 = 0 \end{cases} \Rightarrow \lambda_1 = 2, \quad \lambda_2 = 0, \quad \lambda_3 = 0.$$

Таким чином, $\Lambda^0 = (2{,}0{,}0)$, а $\left(\overline{X}^0, \Lambda^0\right) = (4{,}3{,}2{,}0{,}0)$ - сідлова точка функції Лагранжа.

Перевіримо, що $\left(\overline{X}^0, \Lambda^0\right)$ задовольняє умову (3.16).

$$L\left(\overline{X}, \Lambda^0\right) = -(x_1 - 5)^2 - (x_2 - 1)^2 + 2(-x_1 + 2x_2 - 2) = -x_1^2 + 8x_1 - 16 - x_2^2 +$$
$$+ 6x_2 - 9 - 5 = -5 - (x_1 - 4)^2 - (x_2 - 3)^2 \leq -5$$
$$L\left(\overline{X}^0, \Lambda^0\right) = -(4-5)^2 - (3-1)^2 + 2(-4 + 2 \cdot 3 - 2) = -5$$
$$L\left(\overline{X}^0, \Lambda\right) = -(4-5)^2 - (3-1)^2 + 2(-4 + 2 \cdot 3 - 2) +$$
$$+ \lambda_2(7 \cdot 4 - 2 \cdot 3 - 13) + \lambda_3(-4 - 3 \cdot 3 + 21). = -5 + 9\lambda_2 + 8\lambda_3 \geq -5,$$

тобто $L\left(\overline{X}, \Lambda^0\right) \leq L\left(\overline{X}^0, \Lambda^0\right) \leq L\left(\overline{X}^0, \Lambda\right)$. Умову (3.16) сідлової точки виконано. Таким чином, теорема Куна – Такера підтверджує, що точка $\overline{X}^0(4,3)$ є розв'язком задачі опуклого програмування, яку ми розв'язували.

4 ДИНАМІЧНЕ ПРОГРАМУВАННЯ

У реальному житті є задачі, в яких необхідно враховувати зміни параметрів систем у часі. Ці параметри можуть змінюватися неперервно або дискретно - від етапу до етапу. Наприклад, рік у рік змінюється вік машин та устаткування, виробнича потужність і продуктивність праці на підприємствах, фондовіддача. Необхідно обирати оптимальні рішення на рік (або інший строк) і одночасно на весь розглянутий період у цілому з урахуванням можливих змін параметрів. Показником ефективності можуть служити, наприклад, прибуток, собівартість, сумарні витрати тощо. Для розв'язання такого виду багатокрокових задач розроблений відповідний математичний апарат, який отримав назву *динамічне програмування*.

Метод динамічного програмування був запропонований та розвинений Р. Беллманом і його учнями на початку 50-х років ХХ сторіччя. В основу цього методу покладено принцип оптимальності.

Принцип оптимальності Р. Беллмана. Якщо деяка послідовність розв'язків оптимальна, то окремі наступні розв'язки усередині неї оптимальні стосовно попередніх розв'язків.

Відповідно до цього принципу задачу розкладають на прості етапи та починають розв’язувати з останнього етапу. Потім переходять до попереднього етапу, відкидаючи можливі альтернативи, які суперечать вже відпрацьованому розв’язку.

Одним з найбільш численних класів економічних задач, які розв'язуються методом динамічного програмування, є задача керування запасами. У рамках цієї задачі планується діяльність промислової групи k підприємств на період m років. На початку періоду для розвитку групи виділяються кошти, які потрібно якось розподілити між підприємствами. У процесі роботи підприємств частина коштів, які вкладені в них $x_{i1}, x_{i2}, \dots x_{ik}$ (перший індекс – номер кроку, другий – номер підприємства), витрачається, а частина залишається і може бути знову перерозподілена. Кожне підприємство за рік приносить прибуток, який є заданою функцією вкладених у нього коштів - $f_j(x)$. Ставиться питання: яку кількість коштів потрібно виділити кожному підприємству на початку року, щоб сумарний прибуток за m років був максимальним?

Ця задача є задачею *динамічного програмування*. Розподіл коштів є керований процес, тобто ми можемо обирати деякі параметри, які впливають на прибуток на даному кроці та на сумарний прибуток в цілому. Кошти, які виділяються підприємствам $x_i = \left(x_{i1}, x_{i2}, \dots x_{ik} \right)$, будемо називати вектором *крокового керування*. *Керування* всією операцією $x = \left(x_1, x_2, \dots x_m \right)$ складається із сукупності усіх крокових керувань.

Потрібно знайти *оптимальне керування* x^*, при якому сумарний прибуток максимальний:

$$W = \sum_{ij} f_j(x_{ij}) \to \max .$$

Розв’язання цієї задачі розглянемо на прикладі.

Приклад. Нехай $a = 10000$ од. – кількість капіталовкладень, які необхідно розподілити між двома галузями X та Y.

Кількість капітальних коштів x од., які вкладені у галузь X, приносить прибуток $f_X(x) = 2,2 \cdot x$ од., та наприкінці року залишається $g_X(x) = 0,7 \cdot x$ од.

Кількість капітальних коштів y од., які вкладені у галузь Y, приносить прибуток $f_Y(y) = 2,5 \cdot y$ од. та залишок $g_Y(y) = 0,6 \cdot y$ од.

Наприкінці року залишкові капіталовкладення заново перерозподіляють по галузям. *Потрібно розподілити кошти між галузями так, щоб сумарний прибуток за 3 роки був максимальним.*

Розв'язання. Планування починаємо з останнього року.

Кошти, що залишилися з *к*-го року, позначимо a_k. Кошти, які виділені галузі X в *к*-й рік, позначимо x_k, галузі Y – y_k. Всього на *к*-й рік виділено $a_k = x_k + y_k$. Виразимо y_k

$$y_k = a_k - x_k \tag{4.1}$$

Прибуток к-го року обчислюємо за формулою

$$f_X(x_k) + f_Y(y_k) = 2,2x_k + 2,5y_k = 2,2x_k + 2,5\,\underbrace{(a_k - x_k)}_{\textit{формула }(5.1)} = 2,5a_k - 0,3x_k. \tag{4.2}$$

Залишок коштів *к*-го року обчислюємо за формулою

$$g_X(x_k) + g_Y(y_k) = 0,7x_k + 0,6y_k = 0,7x_k + 0,6\,\underbrace{(a_k - x_k)}_{\textit{формула }(5.1)} = 0,6a_k + 0,1x_k. \tag{4.3}$$

Залишок коштів (*к*-1)-го року буде розподілений на наступний *к*-й рік, тому з іншого боку

$$a_k = g_X(x_{k-1}) + g_Y(y_{k-1}) = \underbrace{0,6a_{k-1} + 0,1x_{k-1}}_{\textit{формула }(5.3)}. \tag{4.4}$$

Максимальний прибуток знаходимо за функцією Беллмана (прибуток за поточний та наступні роки періоду, який розглядається)

$$\mathcal{F}_k(a_k) = \max_{0 \le x_k \le a_k}[f_X(x_k) + f_Y(y_k) + \mathcal{F}_{k+1}(a_k)] \tag{4.5}$$

Зміст функції (4.5) полягає в тому, що потрібно підібрати значення змінної x_k, змінюючи її у вказаних границях від 0 до a_k так, щоб *величина прибутку* за *к*-й та наступні роки була *найбільшою*.

Формула (4.5) є рекурентною формулою, тому $\mathcal{F}_1(a_1)$ – це *максимальний прибуток* за всі роки періоду, що розглядається.

Використовуючи формулу (4.2), перепишемо функцію Беллмана у вигляді

$$\mathcal{F}_k(a_k) = \max_{0 \le x_k \le a_k}[2,5a_k - 0,3x_k + \mathcal{F}_{k+1}(a_k)] \qquad (4.6)$$

Планування починаємо з третього року

***k* = 3**

Оскільки планування включає тільки три роки, будемо вважати, що $\mathcal{F}_4(a_3) = 0$. Тоді з формули (4.6) функція Беллмана має вигляд

$$\mathcal{F}_3(a_3) = \max_{0 \le x_3 \le a_3}[2,5a_3 - 0,3x_3 + \mathcal{F}_4(a_3)] = \underbrace{\max_{0 \le x_3 \le a_3}[2,5a_3 - 0,3x_3]}_{\substack{\textit{треба максимізувати різницю,} \\ \textit{чим менше віднімемо, тим більше} \\ \textit{отримаємо, тому } x_3 = 0}} =$$

$$= 2,5a_3 - 0,3 \cdot 0 = 2,5a_3.$$

Оскільки $x_3 = 0$, то за формулою (4.1) $y_3 = a_3 - 0 = a_3$. Це означає, що кошти у третій рік ми повністю віддамо галузі Y.

***k* = 2**

За формулою (4.6) запишемо функцію Беллмана

$$\mathcal{F}_2(a_2) = \max_{0 \le x_2 \le a_2}[2,5a_2 - 0,3x_2 + \mathcal{F}_3(a_2)].$$

Розглянемо, що таке $\mathcal{F}_3(a_2)$. З попереднього кроку маємо

$$\mathcal{F}_3(a_3) = 2,5a_3 = 2,5\underbrace{(0,6a_2 + 0,1x_2)}_{\textit{формула } (5.4)} = 1,5a_2 + 0,25x_2 = \mathcal{F}_3(a_2).$$

Отже, функція Беллмана для другого року має вигляд

$$\mathcal{F}_2(a_2) = \max_{0\le x_2\le a_2}[2,5a_2 - 0,3x_2 + \mathcal{F}_3(a_2)] =$$
$$= \max_{0\le x_2\le a_2}[2,5a_2 - 0,3x_2 + 1,5a_2 + 0,25x_2] = \underbrace{\max_{0\le x_2\le a_2}[4a_2 - 0,05x_2]}_{\substack{\text{під знаком максимума різниця,} \\ \text{тому } x_2=0}} =$$
$$= 4a_2 - 0,05\cdot 0 = 4a_2.$$

Оскільки $x_2 = 0$, за формулою (5.1) $y_2 = a_2 - 0 = a_2$. Тобто у другий рік всі кошти також віддаємо галузі Y.

***k* = 1**

За формулою (4.6) запишемо функцію Беллмана

$$\mathcal{F}_1(a_1) = \max_{0\le x_1\le a_1}[2,5a_1 - 0,3x_1 + \underset{\substack{\| \\ \mathcal{F}_2(a_2)=4a_2}}{\mathcal{F}_2(a_1)}] = \max_{0\le x_1\le a_1}[2,5a_1 - 0,3x_1 + 4a_2] =$$
$$= \max_{0\le x_1\le a_1}[2,5a_1 - 0,3x_1 + 4\underbrace{(0,6a_1 + 0,1x_1)}_{\text{формула (5.4)}}] =$$
$$= \max_{0\le x_1\le a_1}[2,5a_1 - 0,3x_1 + 2,4a_1 + 0,4x_1] = \underbrace{\max_{0\le x_1\le a_1}[4,9a_1 + 0,1x_1]}_{\substack{\text{треба максимізувати суму, чим} \\ \text{більше додамо, тим більше} \\ \text{отримаємо, тому } x_1=a_1}} =$$
$$= 4,9a_1 + 0,1a_1 = 5a_1 \tag{4.7}$$

Оскільки $x_1 = a_1$, за формулою (4.1) $y_1 = a_1 - a_1 = 0$, тобто у перший рік всі кошти ми віддамо галузі X;
a_1 - це кошти які виділяються на перший рік, за умовою $a_1 = a = 10000$.

Максимальний прибуток, який ми отримаємо за три роки, згідно з (4.7) дорівнює $\mathcal{F}_1(10\,000) = 5\cdot 10\,000 = 50\,000$ од.

Підведемо підсумок, та перевіримо результат

В перший рік виділені кошти $a_1 = 10000$. Ці кошти, вкладені в галузь X ($x_1 = a_1 = 10000$), принесуть *прибуток* $f_X(10000) = 2,2 \cdot 10000 = 22000$. *Залишок* коштів у цей рік становитиме $g_X(10000) = 0,7 \cdot 10000 = 7000$.

У другий рік виділені кошти – це залишок першого року $a_2 = 7000$. Ці кошти, вкладені в галузь Y ($y_2 = a_2 = 7000$), принесуть прибуток $f_Y(7000) = 2,5 \cdot 7000 = 17500$. *Залишок* коштів у другий рік становитиме $g_Y(7000) = 0,6 \cdot 7000 = 4200$.

У третій рік виділені кошти – це залишок другого року $a_3 = 4200$. Ці кошти, вкладені в галузь Y ($y_3 = a_3 = 4200$), принесуть прибуток $f_Y(4200) = 2,5 \cdot 4200 = 10500$. *Залишок* коштів становитиме $g_Y(4200) = 0,6 \cdot 4200 = 2520$.

Ці дані зручно записувати у таблиці розподілу коштів між галузями за три роки.

	Галузь X			Галузь Y		
	Виділені кошти	прибуток	залишок	Виділені кошти	прибуток	залишок
1-й рік	10000	22 000	7 000	–	–	–
2-й рік	–	–	–	7 000	17 500	4 200
3-й рік	–	–	–	4 200	10 500	2 520

Максимальний прибуток згідно з таблицею розподілу коштів буде становити

$$f_X(x_1) + f_Y(y_2) + f_Y(y_3) = 22000 + 17500 + 10500 = \underline{50000 \text{ од.}}$$

Цей результат збігається з одержаним раніше ($\mathcal{F}_1(10000) = 50000$) і є перевіркою правильності розв'язку.

Залишок коштів, оскільки в 3-й рік всі кошти дісталися галузі Y, складає $g_3(y_3) = 0,6 \cdot 4200 = \underline{2520 \text{од.}}$

5 ВАРІАНТИ ІНДИВІДУАЛЬНИХ ЗАВДАНЬ

ЗАВДАННЯ 1 Матрична гра

Розв’язати матричну гру, попередньо спростивши платіжну матрицю

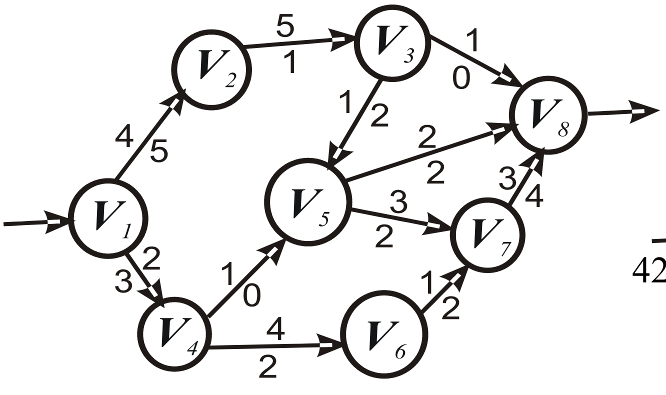
1 $\begin{pmatrix} 0 & -1 & 3 & 2 & -1 \\ 3 & 2 & -1 & -1 & 2 \\ 1 & 0 & 3 & 3 & 0 \\ 1 & 1 & 0 & -1 & 1 \end{pmatrix}$	2 $\begin{pmatrix} -3 & 0 & 2 & -2 & -3 \\ -2 & 1 & 2 & -3 & -3 \\ 1 & -2 & -3 & 1 & 0 \\ 3 & -1 & 0 & 2 & 2 \end{pmatrix}$	3 $\begin{pmatrix} -2 & -2 & 0 & -1 & -2 \\ 0 & 0 & 1 & -1 & 1 \\ -1 & 0 & 2 & -2 & -1 \\ 3 & 3 & 0 & 3 & 0 \end{pmatrix}$
4 $\begin{pmatrix} 2 & 3 & -1 & -2 & -1 \\ -1 & 0 & 2 & 1 & 1 \\ -2 & -1 & 0 & -2 & -2 \\ -1 & 0 & 1 & 1 & 1 \end{pmatrix}$	5 $\begin{pmatrix} 0 & 1 & 4 & 2 & 0 \\ 1 & 2 & 0 & -1 & 0 \\ -1 & -1 & 3 & 1 & -2 \\ 3 & 3 & -1 & -2 & 3 \end{pmatrix}$	6 $\begin{pmatrix} 1 & 0 & 1 & 2 & -1 \\ 0 & 0 & 4 & 4 & 1 \\ 3 & 2 & 1 & 1 & 3 \\ 2 & 1 & -1 & 0 & 2 \end{pmatrix}$
7 $\begin{pmatrix} 0 & 2 & 1 & 0 & -1 \\ -1 & -1 & 0 & -1 & 1 \\ 2 & 3 & 3 & 2 & 0 \\ -1 & 0 & 1 & -1 & 3 \end{pmatrix}$	8 $\begin{pmatrix} -2 & -1 & 0 & -1 & -3 \\ -1 & 0 & -1 & -2 & -3 \\ -2 & -3 & -2 & -3 & -3 \\ -2 & 1 & 3 & 2 & 2 \end{pmatrix}$	9 $\begin{pmatrix} 2 & 1 & 3 & 0 & 3 \\ -2 & 0 & 1 & -1 & -2 \\ 1 & 2 & 0 & 1 & -1 \\ 0 & -1 & -2 & 1 & -2 \end{pmatrix}$
10 $\begin{pmatrix} 3 & -1 & 2 & 3 & -2 \\ 0 & 2 & -1 & 0 & 1 \\ -1 & 1 & -1 & -1 & 0 \\ 2 & -2 & 1 & 2 & -2 \end{pmatrix}$	11 $\begin{pmatrix} -1 & 3 & 0 & 1 & -2 \\ -2 & 0 & -1 & -1 & -2 \\ -1 & -2 & 0 & 1 & -1 \\ 1 & -1 & 2 & 1 & 0 \end{pmatrix}$	12 $\begin{pmatrix} -1 & 0 & 0 & -1 & 1 \\ 2 & 1 & -2 & 3 & -1 \\ -2 & -1 & 0 & -3 & 0 \\ -1 & 0 & -3 & -2 & -1 \end{pmatrix}$
13 $\begin{pmatrix} 2 & 1 & 1 & 3 & 2 \\ 0 & 5 & 4 & 3 & 0 \\ -1 & 2 & 0 & 1 & -1 \\ -1 & 3 & 2 & 0 & 0 \end{pmatrix}$	14 $\begin{pmatrix} 0 & -3 & -3 & -3 & -1 \\ -3 & -1 & -1 & -2 & -2 \\ 2 & -2 & -2 & -3 & 2 \\ -2 & 0 & 0 & -1 & 1 \end{pmatrix}$	15 $\begin{pmatrix} -1 & -2 & 3 & 3 & 0 \\ 1 & 0 & -1 & -2 & 2 \\ 2 & 2 & 0 & 0 & 2 \\ 0 & -2 & 3 & 2 & -1 \end{pmatrix}$

16 $\begin{pmatrix} 2 & 0 & 2 & -2 & -1 \\ 0 & -2 & 0 & -3 & -2 \\ -2 & -1 & -2 & -1 & 0 \\ -1 & 1 & -1 & 0 & 1 \end{pmatrix}$	17 $\begin{pmatrix} 0 & 1 & 0 & -1 & 1 \\ 3 & -1 & -2 & 3 & -1 \\ 2 & -2 & -2 & 0 & 2 \\ 0 & -2 & -2 & -1 & 0 \end{pmatrix}$	18 $\begin{pmatrix} -3 & 0 & -2 & 1 & -2 \\ -3 & 1 & -1 & 2 & -1 \\ 2 & -1 & 2 & 0 & 2 \\ -2 & -3 & -1 & 0 & 0 \end{pmatrix}$
19 $\begin{pmatrix} -2 & -1 & -2 & -1 & -1 \\ 1 & 0 & -1 & 2 & 0 \\ 0 & 4 & 3 & 1 & 4 \\ -1 & 2 & 1 & -2 & 2 \end{pmatrix}$	20 $\begin{pmatrix} -1 & -2 & 2 & 1 & -1 \\ 0 & -2 & 1 & 0 & 0 \\ 1 & 1 & -2 & -2 & 1 \\ 3 & 2 & 0 & -1 & 3 \end{pmatrix}$	21 $\begin{pmatrix} -2 & -1 & 1 & 0 & -3 \\ 3 & 0 & 1 & -1 & 2 \\ 2 & -2 & -3 & -1 & 0 \\ -1 & 2 & 1 & 1 & -3 \end{pmatrix}$
22 $\begin{pmatrix} 0 & 3 & 2 & -1 & 3 \\ 0 & -2 & -3 & 0 & -2 \\ -3 & -2 & -2 & -1 & -2 \\ 1 & -1 & -2 & 0 & -1 \end{pmatrix}$	23 $\begin{pmatrix} -1 & -1 & 2 & 0 & 3 \\ 1 & 3 & 0 & -1 & -1 \\ -1 & 0 & 4 & 1 & 3 \\ 2 & 3 & 1 & 2 & 0 \end{pmatrix}$	24 $\begin{pmatrix} 2 & 1 & -1 & 2 & -2 \\ 1 & 0 & 3 & 0 & 2 \\ 4 & 3 & 0 & 3 & -2 \\ 0 & -1 & 1 & -1 & 0 \end{pmatrix}$
25 $\begin{pmatrix} -2 & -1 & 1 & 0 & -2 \\ -1 & -2 & -2 & -2 & -1 \\ -2 & 3 & -1 & -2 & -2 \\ 2 & 0 & 4 & 2 & 2 \end{pmatrix}$	26 $\begin{pmatrix} -1 & 2 & 0 & 3 & 1 \\ 2 & -1 & 3 & -1 & -2 \\ -2 & 1 & -2 & 0 & -1 \\ -2 & -1 & -2 & 1 & -1 \end{pmatrix}$	27 $\begin{pmatrix} -3 & -2 & 0 & -1 & -1 \\ 1 & 1 & -2 & 2 & -3 \\ -3 & -1 & -1 & -2 & -1 \\ 2 & 2 & 1 & 3 & -2 \end{pmatrix}$
28 $\begin{pmatrix} -1 & -3 & -3 & -1 & -2 \\ 0 & -1 & -2 & 0 & -1 \\ 1 & 2 & 2 & 1 & -1 \\ 1 & 3 & -3 & 2 & 1 \end{pmatrix}$	29 $\begin{pmatrix} -1 & 0 & 1 & -1 & 0 \\ 0 & 3 & 4 & 3 & 1 \\ 2 & -1 & 0 & 0 & 3 \\ 1 & -1 & -1 & 1 & 2 \end{pmatrix}$	30 $\begin{pmatrix} 0 & -1 & 0 & 1 & 0 \\ 1 & -2 & -1 & 0 & 1 \\ -1 & -2 & 0 & -1 & -1 \\ -1 & 3 & -2 & 3 & 4 \end{pmatrix}$

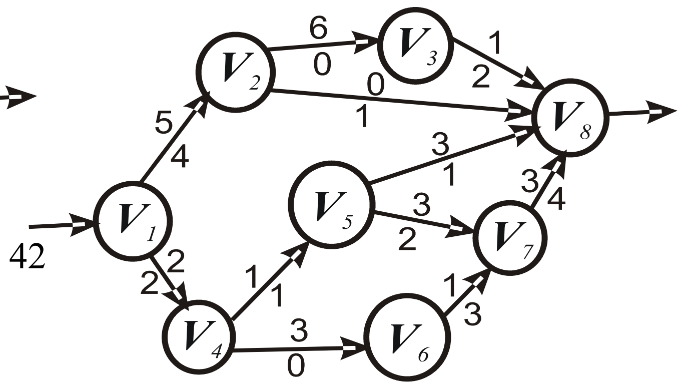
ЗАВДАННЯ 2 Програмування на сітках

Знайти мінімальний розріз та визначити максимальну пропускну спроможність сітки.

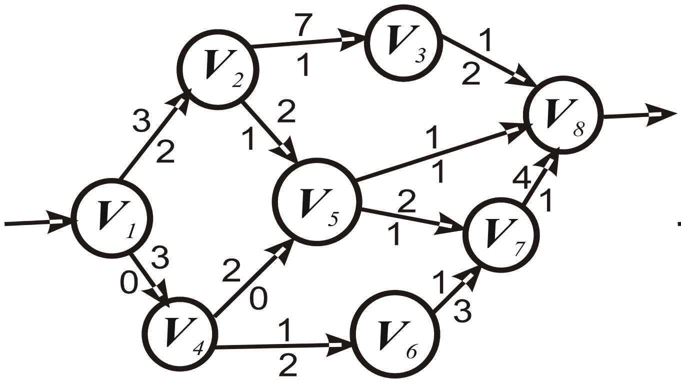
1



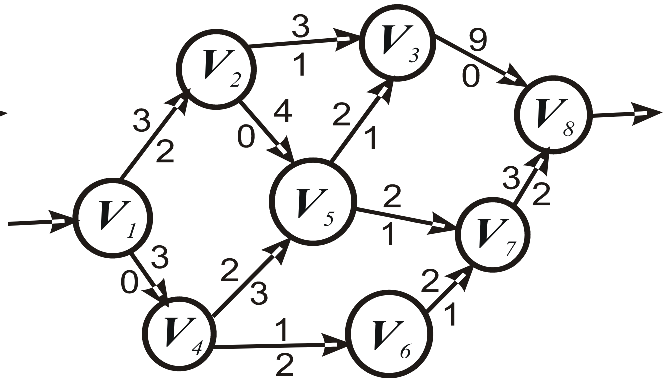
2



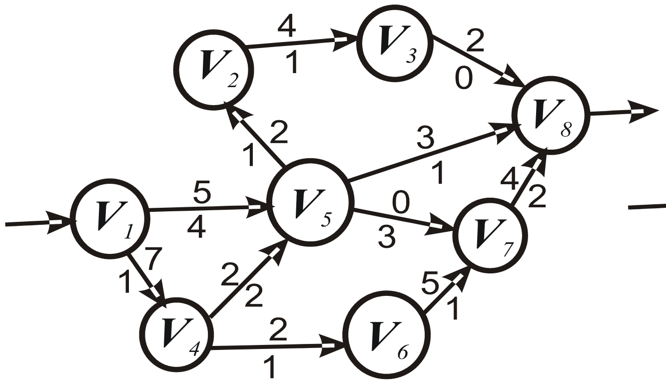
3


V1
V2
V3
V4
V5
V6
V7
V8
7 1
1 2
3 2
2 1
1 1
2 1
4 1
3 0
2 0
1 2
1 3

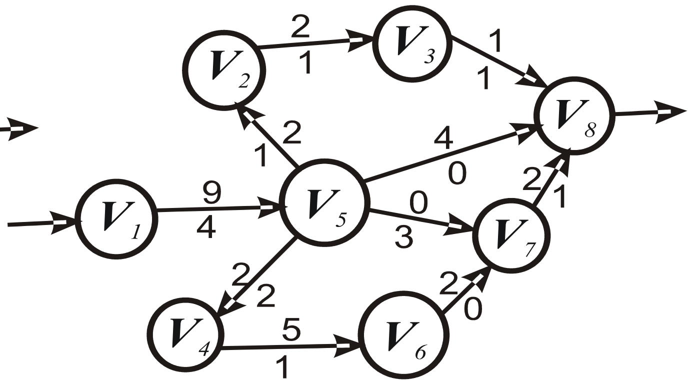
4


V1
V2
V3
V4
V5
V6
V7
V8
3 1
9 0
3 2
4 0
2 1
2 1
3 2
3 0
2 3
1 2
2 1

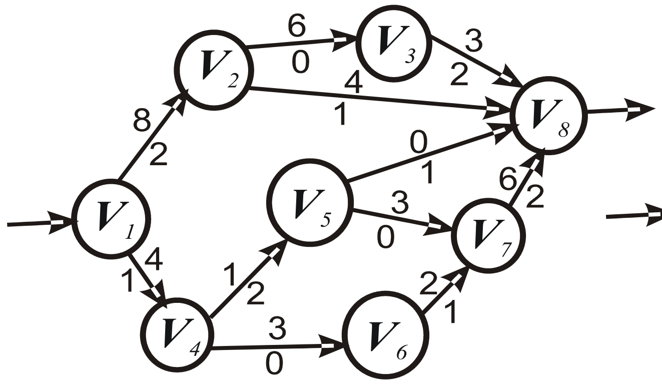
5


V1
V2
V3
V4
V5
V6
V7
V8
4 1
2 0
2 1
3 1
5 4
0 3
4 2
7 1
2 2
2 1
5 1

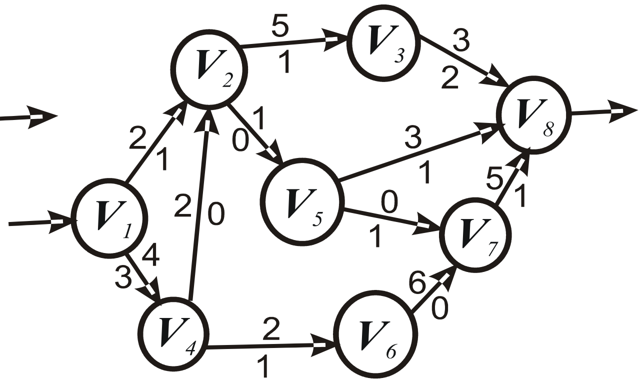
6


V1
V2
V3
V4
V5
V6
V7
V8
2 1
1 1
2 1
4 0
9 4
0 3
2 1
2 2
5 1
2 0

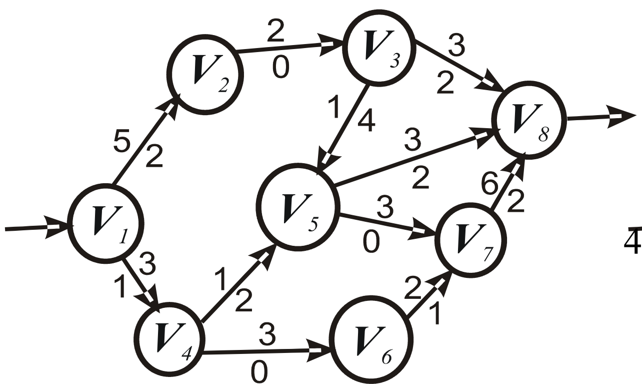
7


V1
V2
V3
V4
V5
V6
V7
V8
6 0
3 2
4 1
8 2
0 1
3 0
6 2
4 1
1 2
3 0
2 1

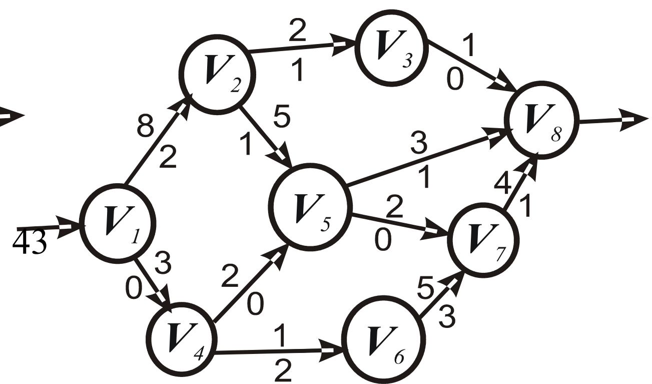
8


V1
V2
V3
V4
V5
V6
V7
V8
5 1
3 2
2 1
1 0
3 1
2 0
0 1
5 1
4 3
2 1
6 0

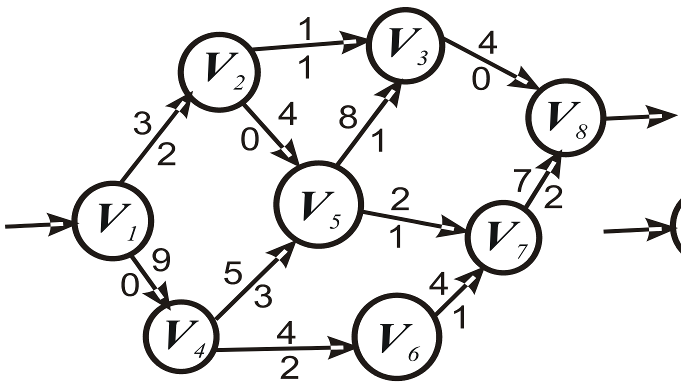
9


V1
V2
V3
V4
V5
V6
V7
V8
2 0
3 2
5 2
1 4
3 2
3 0
6 2
3 1
1 2
3 0
2 1

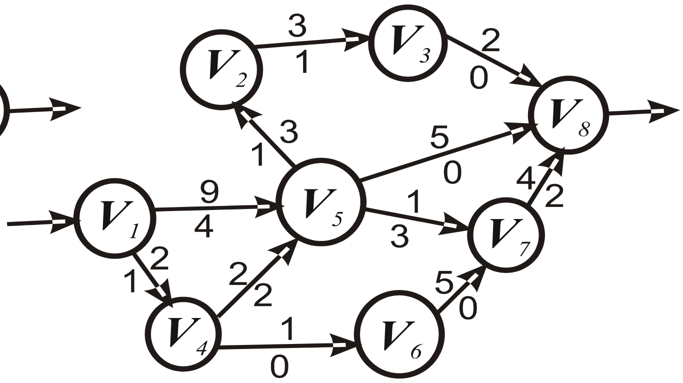
10


V1
V2
V3
V4
V5
V6
V7
V8
2 1
1 0
8 2
5 1
3 1
2 0
4 1
43
3 0
2 0
1 2
5 3

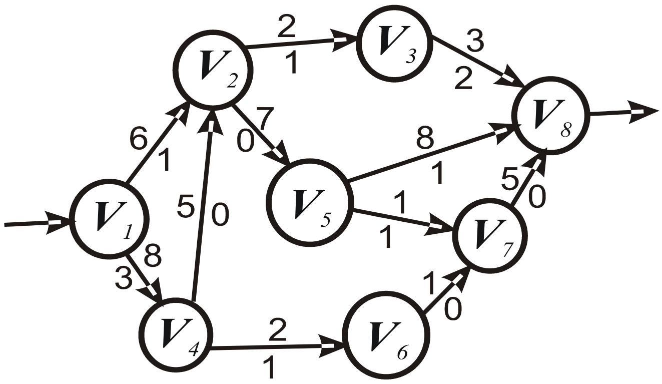
11



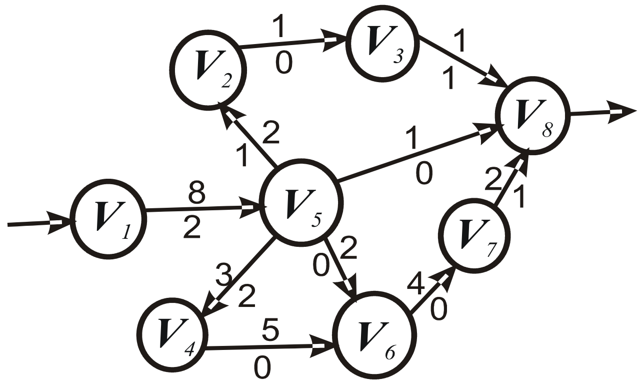
12



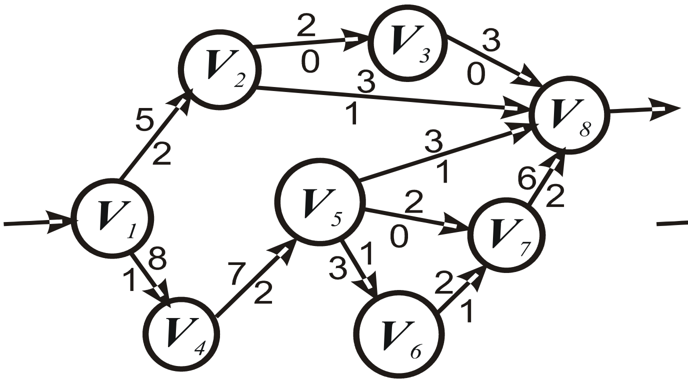
13



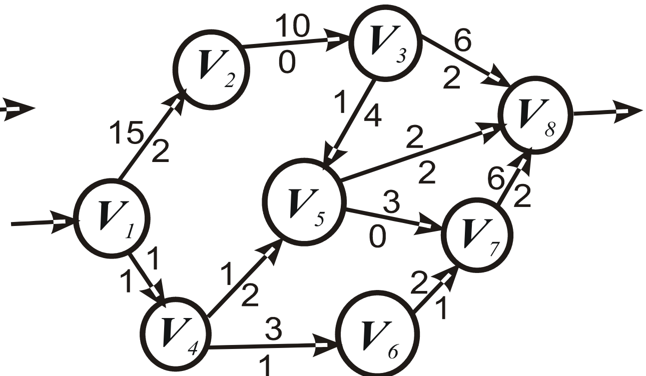
14



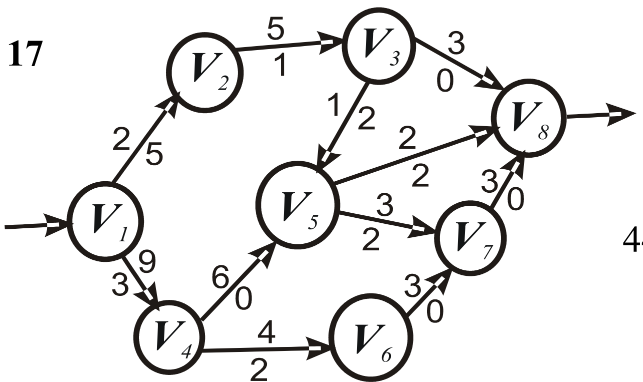
15



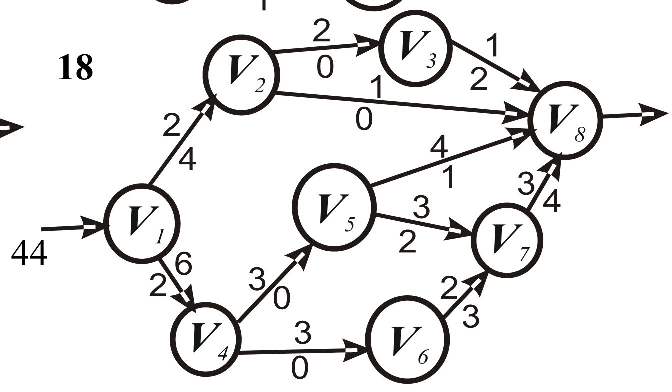
16



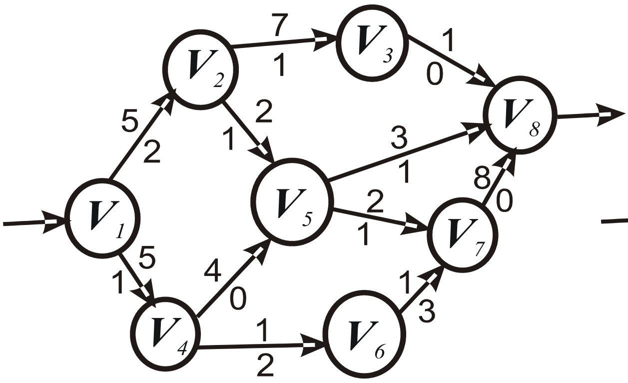
17



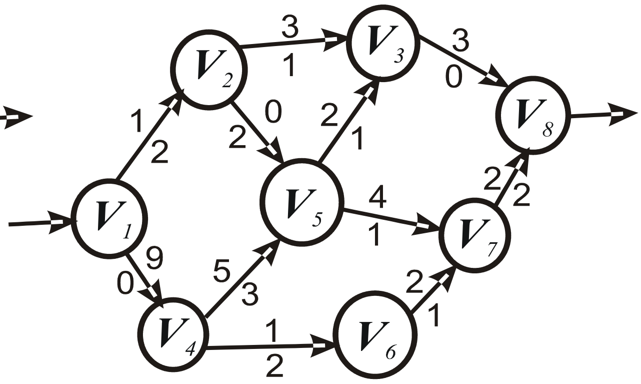
18



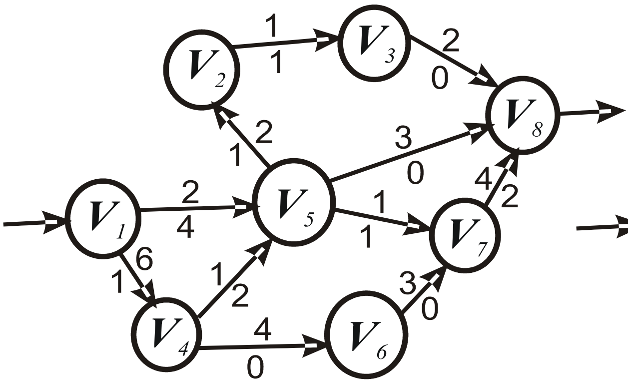
19



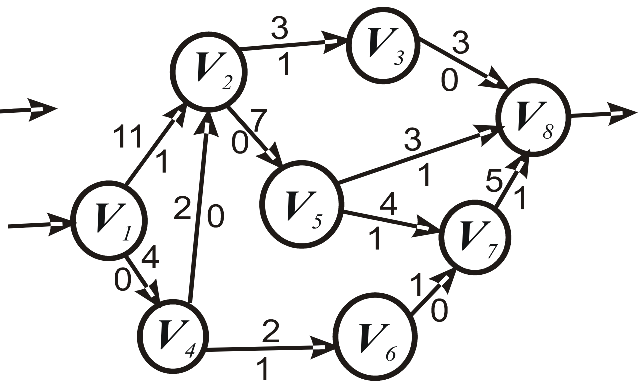
20



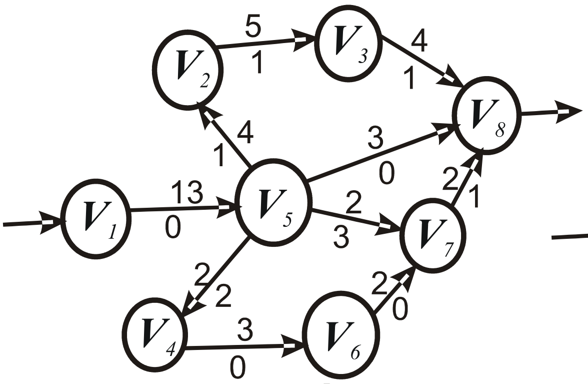
21



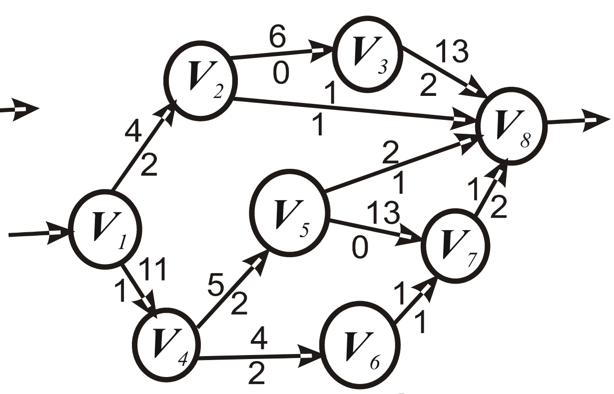
22



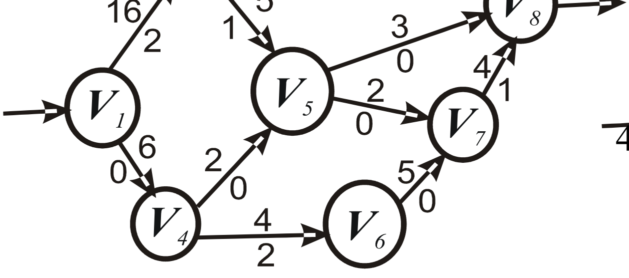
23



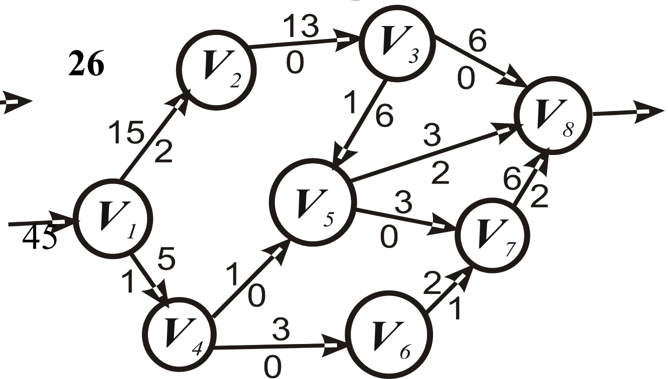
24



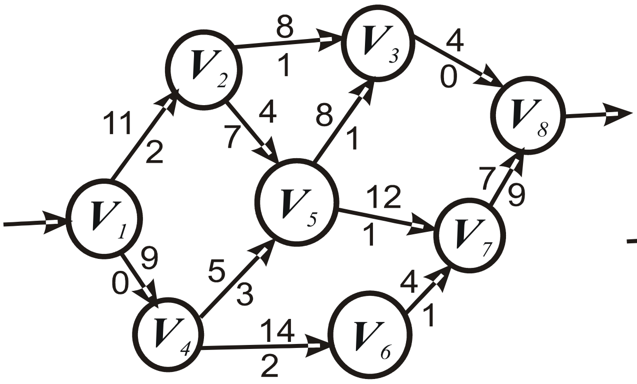
25



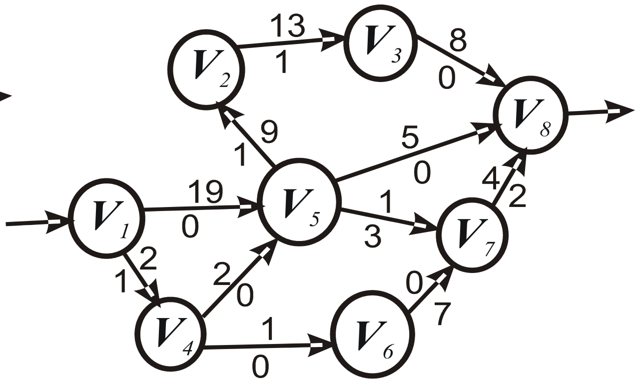
26



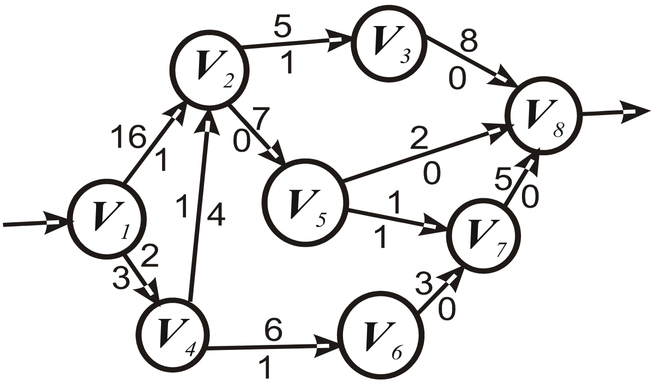
27



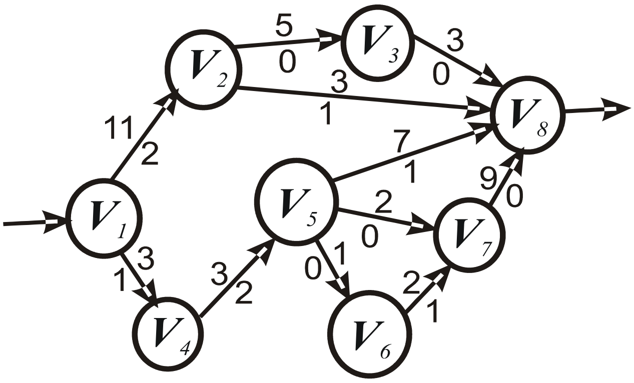
28



29



30



ЗАВДАННЯ 3 Нелінійне програмування

Дано задачу опуклого програмування.

1 Знайти розв’язок задачі графічним способом;

2 Записати функцію Лагранжа даної задачі і знайти її сідлову точку, використовуючи графічний розв’язок задачі.

1) $g=(x_1-2)^2+(x_2-4)^2 \to \min$ $\begin{cases} x_1+x_2\le 8 \\ x_1-2x_2\ge -1 \\ 2x_1-x_2\ge 1,\ \ x_1\ge 0,\ x_2\ge 0 \end{cases}$	2) $f=-(x_1-8)^2-(x_2-1)^2 \to \max$ $\begin{cases} x_1+2x_2\le 13 \\ -x_1+4x_2\ge 5 \\ x_1-x_2\ge 1,\ \ x_1\ge 0,\ x_2\ge 0 \end{cases}$
3) $g=(x_1-7)^2+(x_2-7)^2 \to \min$ $\begin{cases} x_1+x_2\le 4 \\ x_1-x_2\ge 1 \\ 3x_1-x_2\le 9,\ \ x_1\ge 0,\ x_2\ge 0 \end{cases}$	4) $g=(x_1-7)^2+(x_2+14)^2 \to \min$ $\begin{cases} x_1+3x_2\le 24 \\ -3x_1+8x_2\ge 13 \\ 5x_1-2x_2\ge 1,\ \ x_1\ge 0,\ x_2\ge 0 \end{cases}$
5) $g=(x_1-5)^2+(x_2-1)^2 \to \min$ $\begin{cases} 3x_1+4x_2\le 12 \\ x_1+2x_2\ge 1 \\ x_1-x_2\ge -2,\ \ x_1\ge 0,\ x_2\ge 0 \end{cases}$	6) $g=(x_1-12)^2+(x_2-3)^2 \to \min$ $\begin{cases} x_1+x_2\le 11 \\ -x_1+3x_2\ge 1 \\ x_1-x_2\le 1,\ \ x_1\ge 0,\ x_2\ge 0 \end{cases}$
7) $f=-(x_1-6)^2-x_2^2 \to \max$ $\begin{cases} 4x_1+x_2\le 31 \\ x_1-2x_2\le 1 \\ 5x_1-x_2\ge 14,\ \ x_1\ge 0,\ x_2\ge 0 \end{cases}$	8) $f=-(x_1-1)^2-x_2^2 \to \max$ $\begin{cases} x_1+x_2\ge 3 \\ -2x_1+8x_2\le 20 \\ 5x_1-4x_2\le 20,\ \ x_1\ge 0,\ x_2\ge 0 \end{cases}$
9) $g=(x_1-4)^2+x_2^2 \to min$ $\begin{cases} 2x_1+3x_2\le 6 \\ -x_1+2x_2\le 1 \\ -x_1+x_2\ge -1;\ \ x_1\ge 0,\ x_2\ge 0 \end{cases}$	10) $g=(x_1-10)^2+(x_2-1)^2 \to \min$ $\begin{cases} x_1+x_2\ge 1 \\ -3x_1+7x_2\le 21 \\ 4x_1-3x_2\le 12,\ \ x_1\ge 0,\ x_2\ge 0 \end{cases}$
11) $g=(x_1-1)^2+(x_2-2)^2 \to \min$ $\begin{cases} 2x_1+3x_2\ge 6 \\ x_1-x_2\ge 1 \\ 4x_1+5x_2\le 20,\ \ x_1\ge 0,\ x_2\ge 0 \end{cases}$	12) $f=-(x_1+3)^2-(x_2-1)^2 \to \max$ $\begin{cases} x_1+x_2\ge 4 \\ 2x_1-x_2\le 6 \\ -x_1+2x_2\le 2,\ \ x_1\ge 0,\ x_2\ge 0 \end{cases}$

13) $f=-(x_1+2)^2-(x_2+1)^2\to\max$ $\begin{cases}x_1+x_2\ge 5\\ -3x_1+7x_2\le 21\\ 6x_1-4x_2\le 12,\ \ x_1\ge 0,\ x_2\ge 0\end{cases}$	14) $g=(x_1-3)^2+(x_2-4)^2\to\min$ $\begin{cases}2x_1+x_2\le 5\\ -x_1+2x_2\ge 1\\ x_1-x_2\ge -4,\ \ x_1\ge 0,\ x_2\ge 0\end{cases}$
15) $f=-(x_1-7)^2-(x_2-3)^2\to\max$ $\begin{cases}x_1+x_2\le 6\\ x_1\le 5\\ 2x_1+x_2\ge 2,\ \ x_1\ge 0,\ x_2\ge 0\end{cases}$	16) $g=(x_1+1)^2+(x_2-1)^2\to\min$ $\begin{cases}x_1-x_2\le 3\\ 2x_1+x_2\ge 4\\ 5x_1+4x_2\le 20,\ \ x_1\ge 0,\ x_2\ge 0\end{cases}$
17) $f=-(x_1-5)^2-(x_2+1)^2\to\max$ $\begin{cases}6x_1+5x_2\le 30\\ 3x_1-4x_2\ge -12\\ x_1-x_2\le 2,\ \ x_1\ge 0,\ x_2\ge 0\end{cases}$	18) $g=(x_1-5)^2+(x_2-12)^2\to\min$ $\begin{cases}x_1+x_2\ge 4\\ -2x_1+7x_2\le 21\\ 5x_1-3x_2\le 30,\ \ x_1\ge 0,\ x_2\ge 0\end{cases}$
19) $g=(x_1-4)^2+(x_2-5)^2\to\min$ $\begin{cases}x_1+x_2\ge 1\\ x_1-3x_2\ge -3\\ x_1+2x_2\le 4,\ \ x_1\ge 0,\ x_2\ge 0\end{cases}$	20) $g=(x_1-8)^2+(x_2-1)^2\to\min$ $\begin{cases}-x_1+3x_2\le 3\\ 5x_1+4x_2\ge 20\\ 5x_1+6x_2\le 30,\ \ x_1\ge 0,\ x_2\ge 0\end{cases}$
21) $f=-(x_1+1)^2-(x_2-5)^2\to\max$ $\begin{cases}3x_1+x_2\ge 3\\ 3x_1-2x_2\le 6\\ x_1+x_2\le 4,\ x_1\ge 0,\ x_2\ge 0\end{cases}$	22) $g=(x_1-3)^2+(x_2-4)^2\to\min$ $\begin{cases}x_1+2x_2\le 6\\ 3x_1+2x_2\ge 6\\ -x_1+x_2\ge -1,\ \ x_1\ge 0,\ x_2\ge 0\end{cases}$
23) $g=(x_1-5)^2+(x_2-5)^2\to\min$ $\begin{cases}2x_1+x_2\le 5\\ -x_1+2x_2\ge 2\\ -x_1+x_2\le 3,\ \ x_1\ge 0,\ x_2\ge 0\end{cases}$	24) $g=x_1^2+(x_2-4)^2\to\min$ $\begin{cases}x_1+x_2\le 7\\ 3x_1-5x_2\le -3\\ 2x_1-x_2\ge 1,\ \ x_1\ge 0,\ x_2\ge 0\end{cases}$

25) $f=-(x_1-3)^2-(x_2+3)^2\to\max$ $\begin{cases}2x_1-3x_2\ge 6\\3x_1+4x_2\le 24\\x_1+x_2\ge 4,\ x_1\ge 0,\ x_2\ge 0\end{cases}$	26) $g=(x_1-8)^2+(x_2-8)^2\to\min$ $\begin{cases}x_1+x_2\ge 4\\-2x_1+7x_2\ge 21\\5x_1+3x_2\le 30,\ x_1\ge 0,\ x_2\ge 0\end{cases}$
27) $g=(x_1-5)^2+(x_2-8)^2\to\min$ $\begin{cases}-x_1+3x_2\ge 3\\5x_1+3x_2\ge 15\\5x_1+6x_2\le 30,\ x_1\ge 0,\ x_2\ge 0\end{cases}$	28) $g=(x_1+2)^2+(x_2-5)^2\to\min$ $\begin{cases}x_2\le 4\\5x_1-3x_2\ge -15\\x_1-4x_2\le -4,\ x_1\ge 0,\ x_2\ge 0\end{cases}$
29) $f=-(x_1-10)^2-(x_2-6)^2\to\max$ $\begin{cases}5x_1+2x_2\ge 10\\x_1+x_2\le 6\\-x_1+x_2\le 4,\ x_1\ge 0,\ x_2\ge 0\end{cases}$	30) $f=-(x_1-6{,}5)^2-(x_2+2)^2\to\max$ $\begin{cases}2x_1-3x_2\le 6\\3x_1+4x_2\le 24\\x_1+x_2\ge 2,\ x_1\ge 0,\ x_2\ge 0\end{cases}$

ЗАВДАННЯ 4 Задача динамічного програмування

Нехай a од. – кількість капіталовкладень, які необхідно розподілити між двома галузями.

Кількість капітальних коштів x од., що вкладені у галузь X, приносить прибуток $f_X\cdot x$ од., та наприкінці року залишається $g_X\cdot x$ од. Кількість капітальних коштів y од., які вкладені у галузь Y, приносить прибуток $f_Y\cdot y$ од. та залишок $g_Y\cdot y$ од.

Наприкінці року залишкові капіталовкладення заново перерозподіляють між галузями. Потрібно розподілити кошти між галузями так, щоб сумарний прибуток за 3 роки був максимальним.

1	$f_X=1{,}9;\ g_X=0{,}1;$ $f_Y=1{,}4;\ g_Y=0{,}6;$ $a=1000$	2	$f_X=2{,}4;\ g_X=0{,}7;$ $f_Y=2{,}9;\ g_Y=0{,}2;$ $a=2000$	3	$f_X=3{,}9;\ g_X=0{,}3;$ $f_Y=3;\ g_Y=0{,}7;$ $a=3000$
4	$f_X=4{,}1;\ g_X=0{,}9;$ $f_Y=4{,}6;\ g_Y=0{,}4;$ $a=4000$	5	$f_X=5{,}5;\ g_X=0{,}2;$ $f_Y=5;\ g_Y=0{,}7;$ $a=5000$	6	$f_X=6{,}1;\ g_X=0{,}6;$ $f_Y=6{,}6;\ g_Y=0{,}2;$ $a=6000$
7	$f_X=7{,}5;\ g_X=0{,}3;$ $f_Y=7;\ g_Y=0{,}7;$ $a=7000$	8	$f_X=8{,}2;\ g_X=0{,}8;$ $f_Y=8{,}7;\ g_Y=0{,}3;$ $a=8000$	9	$f_x=9{,}5;\ g_X=0{,}4;$ $f_Y=9;\ g_Y=0{,}9;$ $a=9000$

10 $f_X = 1,2;\ g_X = 0,6;$ $f_Y = 1,7;\ g_Y = 0,1;$ $a = 10000$	11 $f_X = 1,6;\ g_X = 0,1;$ $f_Y = 1,1;\ g_Y = 0,6;$ $a = 11000$	12 $f_X = 1,2;\ g_X = 0,7;$ $f_Y = 2;\ g_Y = 0,2;$ $a = 12000$
13 $f_X = 2;\ g_X = 0,3;$ $f_Y = 1,3;\ g_Y = 0,8;$ $a = 13000$	14 $f_X = 1,4;\ g_X = 0,9;$ $f_Y = 2;\ g_Y = 0,4;$ $a = 14000$	15 $f_X = 2;\ g_X = 0,3;$ $f_Y = 1,5;\ g_Y = 0,7;$ $a = 15000$
16 $f_X = 1,6;\ g_X = 0,6;$ $f_Y = 2;\ g_Y = 0,1;$ $a = 16000$	17 $f_X = 2,2;\ g_X = 0,2;$ $f_Y = 1,7;\ g_Y = 0,7;$ $a = 17000$	18 $f_X = 1,8;\ g_X = 0,8;$ $f_Y = 2,3;\ g_Y = 0,3;$ $a = 18000$
19 $f_X = 2,5;\ g_X = 0,4;$ $f_Y = 1,9;\ g_Y = 0,9;$ $a = 19000$	20 $f_X = 2;\ g_X = 0,7;$ $f_Y = 2,6;\ g_Y = 0,2;$ $a = 20000$	21 $f_X = 2,6;\ g_X = 0,3;$ $f_Y = 2,1;\ g_Y = 0,6;$ $a = 21000$
22 $f_X = 2,2;\ g_X = 0,7;$ $f_Y = 2,8;\ g_Y = 0,2;$ $a = 22000$	23 $f_X = 2,9;\ g_X = 0,3;$ $f_Y = 2,3;\ g_Y = 0,8;$ $a = 23000$	24 $f_X = 2,4;\ g_X = 0,9;$ $f_Y = 2,9;\ g_Y = 0,4;$ $a = 24000$
25 $f_X = 2,5;\ g_X = 0,1;$ $f_Y = 2;\ g_Y = 0,8;$ $a = 25000$	26 $f_X = 2,6;\ g_X = 0,6;$ $f_Y = 3;\ g_Y = 0,1;$ $a = 26000$	27 $f_X = 2,7;\ g_X = 0,2;$ $f_Y = 2,1;\ g_Y = 0,7;$ $a = 27000$
28 $f_X = 2,8;\ g_X = 0,8;$ $f_Y = 3,2;\ g_Y = 0,3;$ $a = 28000$	29 $f_1 = 2,9;\ g_1 = 0,4;$ $f_2 = 2,4;\ g_2 = 0,9;$ $a = 29000$	30 $f_1 = 3;\ g_1 = 0,9;$ $f_2 = 3,5;\ g_2 = 0,6;$ $a = 30000$

СПИСОК ЛІТЕРАТУРИ

1 Кузнецов Ю.Н., Кузубов В.И., Волощенко А.Б. Математическое программирование. – М.: Высш. шк., 1986.

2 Кузнецов Ю.Н., Сакович В.А., Холод Н.И. Высш.я математика. Математическое программирование. – Минск, 1994.

3 Вентцель Е.С. Исследование операций. – М.: Сов. радио, 1972.

4 Зуховицкий С.И., Авдеева Л.И. Линейное и выпуклое программирование. – М.: Наука, 1964.

5 Исследование операций в экономике; под ред. Н.Ш. Кремер - М.: ЮНИТИ, 2005.

6 Методичні вказівки і завдання з теми „Математичне програмування” для студентів економічних спеціальностей заочної скороченої форми навчання.– Харьков: УкрДАЗТ, 1999. – Ч. 1, Ч. 2.

7 Математичне програмування. Завдання і методичні вказівки до виконання контрольної роботи.– Харьков: УкрДАЗТ, 2007. – Ч. 1.