

МІЖРЕГІОНАЛЬНА
АКАДЕМІЯ УПРАВЛІННЯ ПЕРСОНАЛОМ



О. Б. Жильцов, В. Р. Кулян, О. О. Юнькова

**МАТЕМАТИЧНЕ
ПРОГРАМУВАННЯ
(з елементами інформаційних технологій)**

За редакцією О. О. Юнькової

*Рекомендовано
Міністерством освіти і науки України
як навчальний посібник для студентів
вищих навчальних закладів*

Київ 2006

ББК 22.18я73
Ж72

Рецензенти: *Ф. Г. Гаращенко*, д-р техн. наук, проф.
В. С. Рогоза, д-р техн. наук, проф.

Схвалено Вченою радою Міжрегіональної Академії управління персоналом (протокол № 7 від 27.08.04)

Рекомендовано Міністерством освіти і науки України (лист № 14/18.2-2765 від 22.12.04)

Жильцов О. Б.

Ж72 Математичне програмування (з елементами інформаційних технологій): Навч. посіб. для студ. вищ. навч. закл. / О. Б. Жильцов, В. Р. Кулян, О. О. Юнькова; За ред. О. О. Юнькової. — К. : МАУП, 2006. — 184 с. : іл. — Бібліогр. : с. 181.

ISBN 966-608-507-0

Навчальний посібник містить необхідний мінімум відомостей із теорії оптимізації. Розглянуто основні оптимізаційні задачі, що виникають у ході прийняття організаційних рішень. Подано традиційні розділи математичного програмування: лінійне та нелінійне програмування, транспортна задача та задачі цілочислового лінійного програмування. Моделювання динаміки економічних процесів викладено на прикладах задач динамічного програмування. Оптимізацію рішень в умовах ризику й невизначеності розглянуто на прикладах задач стохастичного програмування та матричних ігор. Розкрито деякі можливості використання сучасних програмних засобів для розв'язання задач математичного програмування.

Для студентів економічних спеціальностей, що мають певний рівень знань із вищої математики, алгебри та теорії ймовірностей.

ББК 22.18я73

© О. Б. Жильцов, В. Р. Кулян,
О. О. Юнькова, 2006

© Міжрегіональна Академія управління персоналом (МАУП), 2006

ISBN 966-608-507-0

ВСТУП

Для вивчення складних процесів і явищ, коли проведення експериментів потребує значних витрат або взагалі неможливе, застосовують моделювання. *Модель* — це спеціально створений об'єкт, на якому відтворено певні характеристики досліджуваного об'єкта з метою його вивчення, а *моделювання* — спосіб відображення розглянутих характеристик досліджуваного об'єкта.

Моделі можуть бути реалізовані за допомогою фізичних (наприклад, аеродинамічна труба для вивчення обтікання повітрям крила літака чи тренажери для льотчиків і водіїв) й абстрактних об'єктів, описаних засобами штучної мови (наприклад, математичне подання фізичних законів).

Математичне моделювання — найбільш досконалий і ефективний метод моделювання. Природно, що результати дослідження такої моделі мають практичний інтерес, коли вона цілком відповідає (адекватна) розглянутому явищу. Для повнішого опису реальних явищ і процесів доводиться будувати складніші й точніші математичні моделі.

В економічній науці здавна використовують моделі. Одну з перших економічних моделей — модель відтворення Ф. Кене — було створено в 1758 р. Удосконалення економіко-математичних моделей зумовило подальший розвиток моделювання в економіці. Жодна сучасна економічна теорія не обходиться без математичного опису різних економічних явищ і процесів.

Одне з практично найважливіших питань економіки — побудова господарського плану на різних рівнях економічної системи, від цеху до всього господарства країни. Складання найкращого в певному розумінні, або, як кажуть, оптимального, плану — важлива проблема в житті суспільства: одні й ті самі витрати можуть давати різний економічний ефект залежно від прийнятих економічних рішень.

Роботу будь-якого підприємства можна характеризувати певними об'єктивними і технологічними показниками устаткування, які зазвичай не можна змінювати в процесі виробництва. Ці показники називаються некерованими змінними, або параметрами. Ті самі показники, що залежать від суб'єктивних рішень (наприклад, обсяг сировини, запущеної в обробку, або кількість кінцевої продукції, запущеної до випуску), називаються керованими змінними.

Мета виробництва, сформульована у вигляді функціонального співвідношення, що включає в себе як керовані змінні, так і параметри, називається цільовою *функцією*.

Щоб скласти найкращий план господарювання, потрібно оптимізувати цільову функцію, тобто добрати такі значення керованих змінних, для яких цільова функція набуває максимального (якщо мова йде про прибуток) або мінімального (якщо йдеться про собівартість продукції) значення.

Однак неможливо запустити у виробництво більше сировини, ніж є на підприємстві, і не можна випустити більше продукції, ніж дають змогу наявні сировинні, технологічні, фінансові чи інші ресурси. Інакше кажучи, найкращий план слід визначати на обмеженій множині керованих змінних, яка називається *допустимою множиною розв'язків*.

Виражені через керовані змінні цільова функція й обмеження утворюють математичну модель задачі оптимізації. Будь-який набір значень змінних, що задовольняє обмеження, визначає *допустимий план*, а той із них, на якому досягається максимум (мінімум) цільової функції, називається *оптимальним планом*.

Розглянемо деякі задачі планування та керування, математичні моделі яких зводяться до оптимізаційних задач, або так званих задач математичного програмування.

Визначення найкращого складу суміші. Іноді таку задачу називають задачею про вибір дієти. Нехай відомий вміст поживних речовин у різних продуктах харчування, а також калорійність одиниці кожного виду продуктів. Потрібно добрати раціон — набір і кількість продуктів — так, щоб кожна поживна речовина містилася в ньому в потрібній кількості та сумарна калорійність дієти була мінімальною.

До задач про суміші належить і така. Бензини різних марок одержують змішуванням різних нафтопродуктів. При цьому потрібно максимально точно дотримуватися заданих показників якості бензину (октанового числа, ступеня очищення та ін.), але вихідні нафто-

продукти мають різні технічні й економічні характеристики (наприклад, північна та південна нафти істотно різняться за якісним складом), і від того, які нафтопродукти змішати, залежить рентабельність виробництва. У цій задачі потрібно побудувати такий план змішування нафтопродуктів, що забезпечує максимальну рентабельність виробництва та дає змогу одержувати бензини заданих сортів у потрібних обсягах.

Задача про оптимальний план випуску продукції. Нехай підприємство випускає продукцію заданого асортименту. Витрати певного виду ресурсів на випуск одного виробу із зазначеного асортименту, а також повні обсяги наявних ресурсів фіксовані. Прибуток, отримуваний підприємством від виготовлення та реалізації одиниці кожного виду продукції, відомий і постійний. Підприємству потрібно скласти план випуску продукції, який технологічно здійснений за наявних ресурсів усіх видів, задовольняє задані обмеження щодо випуску кожного виду продукції та водночас гарантує найбільший загальний прибуток.

Оптимізація міжгалузевих потоків. Нехай кожна з кількох галузей господарства виробляє тільки один специфічний вид продукції, і кожен вид продукції використовують (зокрема, у нульовій кількості) у виробництві в усіх інших галузях. Потрібно знайти такі можливі в заданих умовах обсяги виробництва кожної галузі й такий план випуску кінцевої продукції, щоб максимізувати загальну вартість виготовленого кінцевого продукту.

Транспортна задача. У найпростішому варіанті вона виникає, коли мова йде про раціональне перевезення якогось однорідного продукту від виробників до споживачів. Передбачено, що споживачам байдуже, звідки, з яких пунктів виробництва надходить продукт, аби його обсяг був належним. Однак від того, наскільки раціонально пункти споживання прикріплено до пунктів виробництва, істотно залежить обсяг транспортної роботи. Тому природно виникає задача про найраціональніше перевезення вантажу, коли потреби задовольняються, а витрати на транспортування мінімальні.

Найпростіша задача розміщення. Нехай у відомих пунктах уже є чи можна розмістити підприємства, що виробляють якийсь продукт, споживаний в інших відомих пунктах. Відомі витрати на виробництво одиниці продукту та можливий максимальний обсяг виробництва в усіх пунктах виробництва, а також витрати на транспортування з пунктів виробництва до пунктів споживання. Потрібно так

розмістити нові підприємства, визначити обсяги виробництва в них і план перевезень, щоб сумарні витрати на виробництво та транспортування всього потрібного обсягу продукту були мінімальні.

Цю задачу можна звести до звичайної транспортної, у якій до витрат на транспортування додано витрати на виробництво в пункті відправлення.

Інші види оптимізаційних задач:

1) *керування запасами* (зі збільшенням запасів збільшуються витрати на збереження, однак зі зменшенням запасів збільшуються витрати через можливу їх нестачу);

2) *розподіл ресурсів* (для певних наборів робіт потрібно так розподілити ресурси, щоб дістати найбільший прибуток від виконання цих робіт або мінімізувати втрати, пов'язані з неповним забезпеченням ресурсами);

3) *ремонт і заміна устаткування* (робоче устаткування з часом зношується, застаріває та підлягає заміні, тому бажано визначити найкращі терміни відновлювального ремонту й момент заміни устаткування модернізованим);

4) *масове обслуговування* (в організаціях, що обслуговують черги замовлень або вимог, наприклад на телефонних станціях, у ремонтних майстернях, білетних касах тощо; завдання полягає в тому, щоб мінімізувати сумарні очікувані втрати від несвоєчасного обслуговування замовлень або простоїв устаткування);

5) *календарне планування* (дає змогу скласти такий розклад завантаження устаткування, щоб сумарна тривалість комплексу завершених робіт була мінімальною);

6) *сіткове планування та керування* (у разі виконання робіт на складних і високовартісних об'єктах, коли потрібно узгоджувати терміни завершення окремих комплексів робіт і моменти запуску операцій усього комплексу);

7) *вибір маршруту* (проєктуючи комунікації чи трубопроводи, потрібно якнайкраще розміщувати їх, щоб оптимізувати потоки в мережах);

8) *комбіновані задачі* (містять водночас кілька типових задач).

Розробка моделей і їх типи. Зупинімося докладніше на формалізації моделей. За ступенем відповідності оригіналу їх поділяють на *ізоморфні*, які строго відповідають оригіналу (їх використовують зазвичай для простих систем), і *гомоморфні*, що відбивають лише певні властивості оригіналу (математичні моделі).

Математичні моделі будують у такій послідовності:

- 1) вивчають і аналізують причинно-наслідкові зв'язки;
- 2) використовують аналогії;
- 3) проводять експерименти (якщо це можливо) для виявлення істотних змінних.

Керовані змінні (значення яких можна змінювати в певних межах) позначимо x_1, x_2, \dots, x_n і об'єднаємо їх у вектор $\mathbf{x} = (x_1, x_2, \dots, x_n)$. Некеровані змінні (параметри), значення яких не можна змінювати, запишемо як вектор $\mathbf{y} = (y_1, y_2, \dots, y_m)$. Розмірності цих векторів можуть бути різними, тобто $n \neq m$.

Нехай функція $E = f(x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_m) = f(\mathbf{x}, \mathbf{y})$, що залежить від керованих і некерованих змінних, — показник якості або ефективності системи.

Набір функцій $g_i(\mathbf{x}, \mathbf{y})$, $i = 1, 2, \dots, m$, визначає співвідношення між показниками функціонування системи, а нерівності $g_i(\mathbf{x}, \mathbf{y}) \leq b$, $i = 1, 2, \dots, m$, задають обмеження на всі наявні в системі ресурси.

Тоді в найзагальнішому вигляді задачу оптимізації можна сформулювати так: *знайти таке значення вектора $\mathbf{x} = (x_1, x_2, \dots, x_n)$, для якого цільова функція $E = f(\mathbf{x}, \mathbf{y})$ досягає екстремального (максимального чи мінімального) значення та при цьому задовольняються всі обмеження на керовані змінні.*

Формально цю задачу можна подати так: знайти

$$\text{extr}E = f(\mathbf{x}, \mathbf{y}) \quad (1)$$

за умови

$$g_i(\mathbf{x}, \mathbf{y}) \leq b, \quad i = 1, 2, \dots, m. \quad (2)$$

Для знаходження оптимального розв'язку задачі (1), (2) застосовують різні методи теорії оптимальних розв'язків, або так званого математичного програмування.

Математичне програмування — розділ математики, що містить теорію та чисельні методи розв'язання багатовимірних екстремальних задач з обмеженнями, тобто задач на екстремум функції багатьох змінних з обмеженнями на область зміни цих змінних. На відміну від класичної теорії екстремальних задач, що являє собою частину математичного аналізу, математичне програмування стосується задач, у яких суттєві обмеження на область зміни змінних.

Залежно від виду цільової функції й обмежень економіко-математичні моделі поділяють на такі види:

1) якщо функції $f(x, y)$ і $g_i(x, y)$, $i = 1, 2, \dots, m$, лінійні відносно x , то маємо **лінійне програмування**;

2) якщо хоча б одна з цих функцій нелінійна відносно x , то маємо задачу **нелінійного програмування**;

3) у **динамічному програмуванні** цільова функція $f(x, y)$ має спеціальну структуру (являє собою суму чи добуток функцій, що залежать від різних аргументів);

4) **стохастичне програмування** застосовують тоді, коли вектор керованих змінних y випадковий (у таких задачах розглядають математичне сподівання цільової функції в разі ймовірнісних обмежень);

5) якщо на керовані змінні накладено обмеження цілісності (змінні x — цілі чи натуральні числа), то застосовують методи **дискретного програмування**;

6) **евристичне програмування** застосовують у тому разі, коли строгими методами не вдається отримати оптимальний розв'язок, а за допомогою евристичних підходів можна знайти розв'язок, близький до оптимального.

У цьому посібнику наведено докладні постановки названих задач і основні методи їх розв'язання.

Контрольні запитання та завдання

1. Що таке математична модель економічних явищ?
2. Означте складові елементи загальної математичної моделі.
3. Що таке керовані змінні моделі?
4. Що таке параметри моделі?
5. Означте складові елементи загальної моделі математичного програмування.
6. Означте поняття допустимої множини розв'язків задачі математичного програмування й опишіть способи її визначення.
7. опишіть способи визначення цільової функції в задачах математичного програмування.
8. Дайте загальну характеристику задач математичного програмування.
9. За якими ознаками класифікують задачі математичного програмування?
10. Зазначте основні класи задач математичного програмування.
11. Наведіть приклади найпростіших економічних задач, для розв'язання яких доцільно застосовувати математичне програмування.
12. Означте предмет математичного програмування.

1. ЛІНІЙНЕ ПРОГРАМУВАННЯ

Розглянемо моделі економічних задач, для розв'язання яких доцільно застосовувати лінійне програмування. Одна з них — задача вибору оптимального асортименту.

Приклад 1.1. Нехай якийсь підприємство виготовляє столи та стільці з двох видів деревини. Для виготовлення одного стола потрібно $0,30 \text{ м}^3$ деревини першого виду та $0,20 \text{ м}^3$ деревини другого виду, а витрати деревини на виготовлення одного стільця становлять відповідно $0,10$ і $0,15 \text{ м}^3$. Запаси кожного виду деревини дорівнюють відповідно 45 і 54 м^3 . Потрібно визначити, скільки слід виготовляти столів і стільців, щоб прибуток підприємства був максимальний, якщо відомо, що прибуток від реалізації одного стола становить 15 грн, а від реалізації одного стільця — 12 грн.

Подамо умову задачі у вигляді таблиці:

Сировина	Вироби		Запас
	Столи	Стільці	
I виду	0,30	0,10	45
II виду	0,20	0,15	54
Прибуток	15	12	—

Якщо позначити як x_1 кількість виготовлених столів, а як x_2 — кількість стільців, то витрати деревини першого виду на виготовлення x_1 столів становлять $0,30x_1 \text{ м}^3$, а на виготовлення x_2 стільців — $0,10x_2 \text{ м}^3$. Ці витрати мають бути не більші, ніж наявні запаси сировини, тобто 45 м^3 , тому виконується нерівність

$$0,30x_1 + 0,10x_2 \leq 45. \quad (1.1)$$

Витрати деревини другого виду на виготовлення x_1 столів становлять $0,20x_1$ м³, а на виготовлення x_2 стільців — $0,15x_2$ м³, і ці витрати не перевищують її запас — 54 м³, тобто виконується нерівність

$$0,20x_1 + 0,15x_2 \leq 54. \quad (1.2)$$

Після реалізації всіх виробів прибуток підприємства становить

$$z = 15x_1 + 12x_2. \quad (1.3)$$

Крім того, на значення x_1 і x_2 накладають так звані *прямі обмеження*:

$$x_1 \geq 0, x_2 \geq 0. \quad (1.4)$$

Вони цілком логічні, адже поняття від'ємної кількості продукції не існує.

Отже, отримуємо математичну модель задачі, що складається з цільової функції z (1.3), обмежень (1.1), (1.2), (1.4) та має такий вигляд: знайти

$$z = 15x_1 + 12x_2 \rightarrow \max$$

для обмежень

$$\begin{cases} 0,30x_1 + 0,10x_2 \leq 45, \\ 0,20x_1 + 0,15x_2 \leq 54, \\ x_1 \geq 0, x_2 \geq 0. \end{cases}$$

Задача збереження матеріалів не менш важлива, ніж попередня. Наприклад, нехай можна виготовляти заготовки з наявної деревини декількома способами, кожен із яких характеризується певною кількістю відходів. Потрібно добрати таку комбінацію способів розрізання деревини, щоб забезпечити комплектність кінцевої продукції й отримати при цьому якомога менше відходів.

Приклад 1.2. Нехай деревина надходить на підприємство у вигляді заготовок, з яких потрібно виготовити деталі двох типів. Кожну заготовку можна розрізати двома способами. У разі розрізання першим способом можна отримати п'ять деталей першого типу та три деталі другого; при цьому залишиться $0,1$ м³ відходів. Розрізаю-

чи заготовки другим способом, можна отримати чотири деталі першого типу та шість деталей другого; при цьому залишиться $0,2 \text{ м}^3$ відходів. Потрібно вибрати, у якій мірі слід застосовувати обидва способи розрізання, щоб виготовити не менше 30 деталей першого типу й не менше 10 деталей другого типу, а також мінімізувати відходи виробництва.

Запишемо початкові умови задачі у вигляді таблиці:

Деталі	Спосіб розрізання		Потрібна кількість деталей
	I	II	
I типу	5	4	30
II типу	3	6	10
Відходи	0,1	0,2	—

Позначимо як x_1 кількість заготовок, розрізаних першим способом, а як x_2 — другим. Тоді загальні відходи від розрізання становлять

$$z = 0,1x_1 + 0,2x_2.$$

Після розрізання першим способом буде отримано $5x_1$ деталей першого типу, а після розрізання другим способом — $4x_2$ деталей того самого типу. Загальна кількість деталей не має бути меншою ніж 30, тобто виконується обмеження

$$5x_1 + 4x_2 \geq 30.$$

Аналогічно, для деталей другого типу після розрізання першим способом буде отримано $3x_1$ деталей, другим — $6x_2$ деталей. Їх має бути не менше ніж 10, тому виконується обмеження

$$3x_1 + 6x_2 \geq 10.$$

Разом із прямими обмеженнями

$$x_1 \geq 0, x_2 \geq 0$$

модель задачі набирає такого вигляду: знайти

$$z = 0,1x_1 + 0,2x_2 \rightarrow \max$$

для обмежень

$$\begin{cases} 5x_1 + 4x_2 \geq 30, \\ 3x_1 + 6x_2 \geq 10, \\ x_1 \geq 0, x_2 \geq 0. \end{cases}$$

У загальному випадку задачу формулюють так.

Нехай підприємство має m видів ресурсів у кількостях b_1, b_2, \dots, b_m і технологічно може випускати n видів різних виробів. Норму витрат ресурсу j -го виду ($j = 1, 2, \dots, m$) на одиницю i -го виробу ($i = 1, 2, \dots, n$) задано, позначимо її як a_{ij} . Ефективність випуску одиниці виробу i -го найменування, тобто прибуток підприємства після його виготовлення та реалізації, — відома величина; вона дорівнює c_i . Потрібно визначити план випуску виробів (оптимальний асортимент), за якого сумарний показник ефективності набуває найбільшого значення.

Нехай x_i — кількість одиниць виробу i -го виду. У цій задачі змінні x_i керовані, а всі інші — фіксовані й заздалегідь задані, тобто некеровані.

Функцію цілі, задану у вигляді

$$E = f(x) = c_1x_1 + c_2x_2 + \dots + c_nx_n = \sum_{i=1}^n c_i x_i,$$

потрібно максимізувати, якнайкраще вибравши значення x_1, x_2, \dots, x_n . Вибір слід робити так, щоб не було перевитрат наявних ресурсів, тобто щоб виконувались обмеження

$$\sum_{i=1}^n a_{ij}x_i \leq b_j, \quad j = 1, 2, \dots, m.$$

Цілком природні в цій задачі й так звані прямі обмеження на керовані змінні, оскільки безглуздо говорити про від'ємні обсяги випуску продукції:

$$x_1 \geq 0, x_2 \geq 0, \dots, x_n \geq 0.$$

Отже, остаточно задача набирає такого вигляду: знайти

$$\max \left(z = \sum_{i=1}^n c_i x_i \right) \quad (1.5)$$

для обмежень

$$\sum a_{ij} x_i \leq b_j, \quad j = 1, 2, \dots, m; \quad (1.6)$$

$$x_1 \geq 0, x_2 \geq 0, \dots, x_n \geq 0. \quad (1.7)$$

У цьому разі цільова функція $f(x) = \sum_{i=1}^n c_i x_i$ й обмеження $\sum_{i=1}^n a_{ij} x_i \leq b_j$ лінійні щодо керованих змінних (x_1, x_2, \dots, x_n уходять у зазначені вирази в першому степені зі сталими коефіцієнтами), тому задача (1.5)–(1.7) називається **задачею лінійного програмування (ЗЛП)**.

Об'єднавши задачі обох типів (максимізації та мінімізації), у загальному вигляді сформулюємо поставлену ЗЛП так: потрібно знайти такі невід'ємні значення $x_1^0, x_2^0, \dots, x_n^0$ змінних x_1, x_2, \dots, x_n , які задовольняють співвідношенням

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n & R_1 & b_1, \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n & R_2 & b_2, \\ \dots & \dots & \dots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n & R_m & b_m, \end{cases} \quad (1.8)$$

й обчислити екстремальне (найбільше чи найменше) значення функції

$$f(x_1, x_2, \dots, x_n) = c_1 x_1 + c_2 x_2 + \dots + c_n x_n \quad (1.9)$$

порівняно з її значеннями для всіх інших наборів змінних x_1, x_2, \dots, x_n , що задовольняють ту саму систему обмежень. Тут R_1, R_2, \dots, R_m — один зі знаків “ \leq ”, “ $=$ ” або “ \geq ”; a_{ij}, b_j, c_i — задані дійсні числа.

Будь-який упорядкований набір (x_1, x_2, \dots, x_n) значень змінних, що задовольняє всі обмеження (1.8) (тобто будь-який розв'язок системи лінійних рівнянь або нерівностей (1.8)), називається **допустимим розв'язком (планом)** задачі. Множина всіх допустимих розв'язків називається **допустимою множиною** задачі. Допустимий розв'язок

$x_1^0, x_2^0, \dots, x_n^0$, для якого цільова функція (1.5) набуває максимального (мінімального) значення, називається *оптимальним розв'язком*, *оптимальним планом* (просто *розв'язком*) розглянутої ЗЛП.

Форми запису задач лінійного програмування

Залежно від обмежень вихідної задачі розрізняють кілька форм подання ЗЛП.

Якщо обмеження задачі записано у вигляді (1.8), а цільову функцію — у вигляді (1.9), то маємо *координатну* форму запису ЗЛП.

Позначимо як c вектор коефіцієнтів при невідомих у цільовій функції, x — вектор керованих змінних, A_j — m -вимірний вектор-стовпець коефіцієнтів при x_j , b — m -вимірний вектор-стовпець вільних членів, що стоять у правій частині рівняння, тобто

$$c = \begin{pmatrix} c_1 \\ c_2 \\ \dots \\ c_n \end{pmatrix}, \quad x = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix}, \quad A_j = \begin{pmatrix} a_{1j} \\ a_{2j} \\ \dots \\ a_{mj} \end{pmatrix}, \quad j = 1, 2, \dots, m, \quad b = \begin{pmatrix} b_1 \\ b_2 \\ \dots \\ b_m \end{pmatrix}.$$

Тоді отримаємо *векторний* запис ЗЛП

$$\begin{aligned} (c, x) &\rightarrow \max, \\ A_1 x_1 + A_2 x_2 + \dots + A_n x_n &R b, \\ x &\geq 0, \end{aligned}$$

де R — знак нерівності. Така форма прийнятна в тому разі, коли обмеження мають однаковий знак (“ \leq ” чи “ \geq ”), а прямі обмеження накладено на всі керовані змінні. Якщо об'єднати всі вектори A_j , $j = 1, 2, \dots, n$, у матрицю $A = (A_1, A_2, \dots, A_n)$, то задача набуде *матрично-векторного* вигляду

$$\begin{aligned} (c, x) &\rightarrow \max, \\ Ax &\leq b, \\ x &\geq 0. \end{aligned}$$

Як і в попередньому записі, замість знака нерівності “ \leq ” в обмеженнях задачі може бути знак “ \geq ”. Прямі обмеження на змінні зберігаються.

$$\begin{cases} a_{k+1,1}x_1 + a_{k+1,2}x_2 + \dots + a_{k+1,n}x_n + x_{n+1} = b_{k+1}, \\ a_{k+2,1}x_1 + a_{k+2,2}x_2 + \dots + a_{k+2,n}x_n + x_{n+2} = b_{k+2}, \\ \dots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n + x_{n+(m-k)} = b_m. \end{cases}$$

Якщо нерівності (1.10) мають протилежний знак, то додаткові змінні віднімаються.

Доведено (зокрема, у [4]), що будь-якому невід'ємному розв'язку $(x_1^0, x_2^0, \dots, x_n^0)$ системи нерівностей відповідає невід'ємний розв'язок системи рівнянь $(x_1^0, x_2^0, \dots, x_n^0, x_{n+1}^0, \dots, x_{n+(m-k)}^0)$.

Змінні, для яких непотрібна умова невід'ємності, формально заміняють різницею двох додатних змінних:

$$x_{r+1} = x'_{r+1} - x''_{r+1}, \quad x_{r+2} = x'_{r+2} - x''_{r+2}, \dots, \quad x_n = x'_n - x''_n,$$

де $x'_{r+1} \geq 0, x''_{r+1} \geq 0, \dots, x'_n \geq 0, x''_n \geq 0$. Очевидно, що така заміна не впливає на попередній результат.

Приклад 1.3. Звести до стандартного вигляду задачу

$$\begin{aligned} & 2x_1 - 3x_2 + 5x_3 - x_4 \rightarrow \max, \\ & \begin{cases} x_1 + 2x_2 - x_3 + x_4 \leq 2, \\ 2x_1 - 3x_2 - 2x_3 + 5x_4 = 4, \\ 3x_1 + x_2 + 3x_3 - x_4 \geq 3, \\ x_1 \geq 0, \quad x_3 \geq 0. \end{cases} \end{aligned}$$

Уводимо дві додаткові змінні $x_5 \geq 0$ й $x_6 \geq 0$ в першому та третьому обмеженнях, а змінні x_2 та x_4 , для яких непотрібна невід'ємність, заміняємо двома різницями: $x_2 = x'_2 - x''_2$; $x_4 = x'_4 - x''_4$, де $x'_2 \geq 0, x''_2 \geq 0, x'_4 \geq 0, x''_4 \geq 0$.

Отже, одержуємо таку задачу в стандартному вигляді:

$$\begin{aligned} & 2x_1 - 3(x'_2 - x''_2) + 5x_3 - (x'_4 - x''_4) + 0 \cdot x_5 + 0 \cdot x_6 \rightarrow \max, \\ & \begin{cases} x_1 + 2(x'_2 - x''_2) - x_3 + (x'_4 - x''_4) + x_5 = 2, \\ 2x_1 - 3(x'_2 - x''_2) - 2x_3 + 5(x'_4 - x''_4) = 4, \\ 3x_1 + (x'_2 - x''_2) + 3x_3 - (x'_4 - x''_4) - x_6 = 3, \\ x_1 \geq 0; x'_2 \geq 0; x''_2 \geq 0; x_3 \geq 0; x'_4 \geq 0; x''_4 \geq 0; x_5 \geq 0; x_6 \geq 0. \end{cases} \end{aligned}$$

У лінійному програмуванні (ЛП) розглядають задачі як максимізації, так і мінімізації цільової функції, однак створювати методи розв'язання для обох задач недоцільно. Простіше замінити задачу максимізації еквівалентною їй задачею мінімізації чи навпаки. Для цього змінюють знак цільової функції, залишаючи незмінними обмеження вихідної ЗЛП, тобто

$$\max f(x_1, x_2, \dots, x_n) = \min [-f(x_1, x_2, \dots, x_n)]$$

і, навпаки,

$$\min f(x_1, x_2, \dots, x_n) = \max [-f(x_1, x_2, \dots, x_n)].$$

Геометрична інтерпретація задачі лінійного програмування

У найпростішому випадку, коли ЗЛП містить тільки дві змінні, легко отримати її геометричну інтерпретацію та розв'язати задачу графічно. Зрозуміло, що цей випадок не має особливого практичного значення, однак він ілюструє основні властивості ЗЛП.

Нехай ефективність виробництва виробів найменувань 1 і 2 становить відповідно $c_1 = 2$; $c_2 = 5$. Підприємство має три види ресурсів у обсягах $b_1 = 400$; $b_2 = 300$; $b_3 = 500$, причому технологія виробництва така, що ресурси першого виду витрачають на виробництво виробів першого найменування, другого — на виготовлення виробів другого найменування, третього — на виготовлення виробів обох найменувань, до того ж у рівних частинах. Модель задачі така: знайти $\max (z = 2x_1 + 5x_2)$ для $x_1 \leq 400$, $x_2 \leq 300$, $x_1 + x_2 \leq 500$ і, природно, $x_1 \geq 0$, $x_2 \geq 0$.

Кожне обмеження-нерівність задає певну півплощину, на межі якої воно виконується як рівність. У цьому прикладі першу півплощину утворюють усі можливі значення (x_1, x_2) , для яких $x_1 \leq 400$, тобто її межа — пряма $x_1 = 400$. Другу півплощину задає пряма $x_2 = 300$; вона містить усі точки (x_1, x_2) , для яких $x_2 \leq 300$. Третю півплощину задає пряма $x_1 + x_2 = 500$; вона містить усі точки (x_1, x_2) , для яких $x_1 + x_2 \leq 500$. Допустима множина розв'язків задачі являє собою перетин усіх трьох півплощин, тобто це множина всіх (x_1, x_2) , для яких виконуються всі три обмеження. Оскільки нас цікавлять лише невід'ємні значення змінних x_1, x_2 , то допустима множина

обмежена ще й координатними осями в першій чверті, тобто для $x_1, x_2 \geq 0$.

Побудувавши на площині x_1Ox_2 відповідні прямі, визначимо перетин усіх трьох півплощин і з урахуванням прямих обмежень на змінні одержимо такий графік (рис. 1.1).

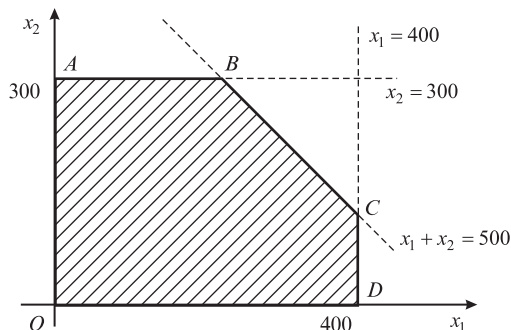


Рис. 1.1

Отже, допустима множина розв'язків даної ЗЛП являє собою багатокутник $OABCD$. Цільова функція має вигляд $f(x_1, x_2) = 2x_1 + 5x_2$. Серед різних наборів значень змінних x_1 і x_2 можна добрати такі, що $2x_1 + 5x_2 = \text{const}$. Вони утворюють множину прямих паралельних ліній, що відповідають різним значенням const , — ліній рівня цільової функції. Цього можна досягти паралельним перенесенням лінії $2x_1 + 5x_2 = 0$ в бік зростання її значень. Як відомо, напрямком зростання довільної функції показує її вектор-градієнт, утворений із перших похідних цієї функції. Для прямої лінії вектор-градієнт — це вектор-нормаль, тобто вектор, перпендикулярний до прямої. Оскільки його можна провести через будь-яку точку площини, то, очевидно, зручніше початок його змістити в початок координат і визначити нормаль до лінії рівня цільової функції, у якій $\text{const} = 0$. У цьому прикладі вектор-нормаль, чи вектор-градієнт, можна задати у вигляді $c = (c_1, c_2) = (2, 5)$, бо

$$\frac{\partial z(x_1, x_2)}{\partial x_1} = c_1 = 2; \quad \frac{\partial z(x_1, x_2)}{\partial x_2} = c_2 = 5.$$

Розглянемо прямі $2x_1 + 5x_2 = z_1$, $2x_1 + 5x_2 = z_2$ та $2x_1 + 5x_2 = z_3$ (рис. 1.2). Очевидно, що виконується співвідношення $z_1 < z_2 < z_3$, тобто пряма, що проходить через точку B , найкраща, бо для всіх ліній, розміщених нижче, $z_1 < z_2$. Третя пряма хоч і відповідає більшій константі $z_3 > z_2$, але не має з допустимою множиною розв'язків спільних точок. Очевидно, що пряма, яка проходить через точку B , — шукана.

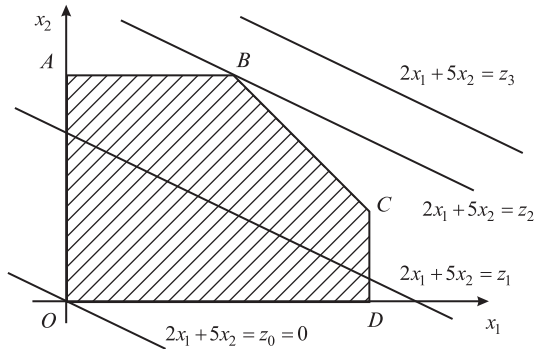


Рис. 1.2

Отже, єдина точка, що належить водночас прямій $f(x_1, x_2) = z_2$ та допустимій множині розв'язків, тобто точка B — оптимальний розв'язок поставленої ЗЛП.

Координати цієї точки $(200; 300)$ легко визначити на графіку.

Незважаючи на обмеженість графічного методу, він може бути корисним і для задач із більшою розмірністю, а саме, коли обмеження задачі записано у вигляді рівнянь, а різниця між кількістю змінних і обмежень задачі дорівнює двом ($n - m = 2$), тобто рівняння містять по дві вільні невідомі.

Властивості розв'язків задач лінійного програмування

За такої геометричної інтерпретації стають очевидними основні властивості ЗЛП, які можна сформулювати у вигляді теорем. Попередньо введемо кілька означень.

Означення 1.1. Систему векторів A_1, A_2, \dots, A_n називають **лінійно залежною**, якщо серед них існує вектор A_k , який лінійно виражається через інші вектори системи, тобто якщо існують такі дійсні числа α_i , що

$$A_k = \alpha_1 A_1 + \dots + \alpha_{k-1} A_{k-1} + \alpha_{k+1} A_{k+1} + \dots + \alpha_n A_n.$$

Якщо жоден із векторів системи лінійно не виражається через решту, то таку систему називають **лінійно незалежною**.

Зауваження. Вектори A_1, A_2, \dots, A_m називаються **лінійно незалежними**, якщо рівність $\alpha_1 A_1 + \alpha_2 A_2 + \dots + \alpha_m A_m = 0$ можлива тільки в разі $\alpha_1 = \alpha_2 = \dots = \alpha_m = 0$.

Означення 1.2. Лінійний простір називають **n -вимірним**, якщо в ньому існує n лінійно незалежних векторів. Систему лінійно незалежних векторів A_1, A_2, \dots, A_n називають **базисом** n -вимірного простору R^n .

Приклад 1.4. Розглянемо набір векторів

$$A_1 = \begin{pmatrix} 1 \\ 0 \\ 3 \end{pmatrix}; \quad A_2 = \begin{pmatrix} 2 \\ 2 \\ 1 \end{pmatrix}; \quad A_3 = \begin{pmatrix} -4 \\ -1 \\ 3 \end{pmatrix}; \quad A_4 = \begin{pmatrix} 3 \\ -1 \\ 0 \end{pmatrix}; \quad A_5 = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}.$$

Відомо, що в тривимірному просторі базисними можуть бути тільки три вектори. Візьмемо вектори A_2, A_3, A_5 і запишемо їх у вигляді матриці $B = (A_2, A_3, A_5)$. Легко переконатися, що матриця B має відмінні від нуля визначники аж до третього порядку, тобто складається з лінійно незалежних векторів, які можна вважати базисними. Вона називається **базисною**.

Теорема 1.1. Будь-який вектор n -вимірного простору R^n можна подати, до того ж єдиним способом, у вигляді лінійної комбінації базисних векторів.

Рівність $A = \alpha_1 A_1 + \alpha_2 A_2 + \dots + \alpha_n A_n$ називають **розкладом** вектора A за базисом A_1, A_2, \dots, A_n , а числа $\alpha_1, \alpha_2, \dots, \alpha_n$ — **координатами** вектора A відносно цього базиса.

Наприклад, якщо базисна матриця одинична, тобто

$$B = E = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{pmatrix},$$

то довільний вектор $A = (\alpha_1, \alpha_2, \dots, \alpha_n)$ можна записати у вигляді

$$A = \alpha_1 \begin{pmatrix} 1 \\ 0 \\ \dots \\ 0 \end{pmatrix} + \alpha_2 \begin{pmatrix} 0 \\ 1 \\ \dots \\ 0 \end{pmatrix} + \dots + \alpha_n \begin{pmatrix} 0 \\ 0 \\ \dots \\ 1 \end{pmatrix}.$$

Перехід до нового базису. Нехай у просторі R^n є два базиси: старий A_1, A_2, \dots, A_n і новий $A_1^*, A_2^*, \dots, A_n^*$. Згідно з теоремою 1.1 кожен із векторів нового базису можна подати як лінійну комбінацію векторів старого базису:

$$\begin{aligned} A_1^* &= \alpha_{11}A_1 + \alpha_{12}A_2 + \dots + \alpha_{1n}A_n, \\ A_2^* &= \alpha_{21}A_1 + \alpha_{22}A_2 + \dots + \alpha_{2n}A_n, \\ &\dots\dots\dots \\ A_n^* &= \alpha_{n1}A_1 + \alpha_{n2}A_2 + \dots + \alpha_{nn}A_n. \end{aligned}$$

Із цих співвідношень випливає, що перехід від одного базису до іншого визначається матрицею переходу $A = (\alpha_{ij})$ ($i, j = 1, 2, \dots, n$). Ця матриця неособлива, тобто має відмінний від нуля визначник, а не то її рядки й, відповідно, базисні вектори були б лінійно залежні. Зворотний перехід від нового базису до старого виконується за допомогою оберненої матриці A^{-1} .

Нехай $x = (x_1, x_2, \dots, x_n)$ — довільний вектор простору R^n , координати якого задано в старому базисі, а $x^* = (x_1^*, x_2^*, \dots, x_n^*)$ — вектор, елементи якого являють собою координати цього самого вектора в новому базисі. Тоді залежність між ними визначається формулами

$$\begin{pmatrix} x_1^* \\ x_2^* \\ \dots \\ x_n^* \end{pmatrix} = (A^{-1})^T \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix} \quad \text{або} \quad \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix} = A^T \begin{pmatrix} x_1^* \\ x_2^* \\ \dots \\ x_n^* \end{pmatrix},$$

де $A^T, (A^{-1})^T$ — матриці, транспоновані відповідно до A й A^{-1} .

Приклад 1.5. Нехай $\begin{pmatrix} a_{1l} \\ a_{2l} \\ \dots \\ a_{nl} \end{pmatrix} = A_l$ — будь-який небазисний вектор із

розглянутих у попередньому прикладі, причому його координати задано в одиничному базисі; $B = \{A_2, A_3, A_5\}$ — базисна матриця нового базису.

Тоді координати будь-якого вектора в новому базисі можна визначити із системи рівнянь $Bx_l = a_l$:

$$\begin{pmatrix} x_{1l} \\ x_{2l} \\ \dots \\ x_{nl} \end{pmatrix} = B^{-1} \begin{pmatrix} a_{1l} \\ a_{2l} \\ \dots \\ a_{nl} \end{pmatrix},$$

де $\begin{pmatrix} x_{1l} \\ x_{2l} \\ \dots \\ x_{nl} \end{pmatrix}$ — вектор-стовпець шуканих координат, тобто

$$A_l = A_2x_{1l} + A_3x_{2l} + A_5x_{3l}.$$

Тут $B = \begin{pmatrix} 2 & -4 & 1 \\ 2 & -1 & 1 \\ 1 & 3 & 1 \end{pmatrix}$, а обернена до неї матриця $B^{-1} = \frac{1}{3} \begin{pmatrix} -4 & 7 & -3 \\ -1 & 1 & 0 \\ 7 & -10 & 6 \end{pmatrix}$.

Визначимо, наприклад, координати вектора A_1 у новому базисі:

$$\begin{pmatrix} x_{11} \\ x_{21} \\ x_{31} \end{pmatrix} = B^{-1}A_1 = \frac{1}{3} \begin{pmatrix} -4 & 7 & -3 \\ -1 & 1 & 0 \\ 7 & -10 & 6 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 3 \end{pmatrix} = \frac{1}{3} \begin{pmatrix} -13 \\ -1 \\ 25 \end{pmatrix}.$$

Аналогічно, координати A_4

$$\begin{pmatrix} x_{14} \\ x_{24} \\ x_{34} \end{pmatrix} = B^{-1}A_4 = \frac{1}{3} \begin{pmatrix} -4 & 7 & -3 \\ -1 & 1 & 0 \\ 7 & -10 & 6 \end{pmatrix} \begin{pmatrix} 3 \\ -1 \\ 0 \end{pmatrix} = \frac{1}{3} \begin{pmatrix} -19 \\ -4 \\ 31 \end{pmatrix}.$$

Координати базисних векторів не обчислюють, бо вони одиничні:

$$\begin{pmatrix} x_{12} \\ x_{22} \\ x_{32} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}; \quad \begin{pmatrix} x_{13} \\ x_{23} \\ x_{33} \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}; \quad \begin{pmatrix} x_{15} \\ x_{25} \\ x_{35} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}.$$

Якщо у вихідному базисі A_3 замінити вектором A_1 , то лінійна незалежність набору збережеться, тобто $\{A_2, A_1, A_5\}$ — також базис, але координати небазисних векторів у разі розкладу за новим базисом, природно, зміняться.

Означення 1.3. *Опуклою комбінацією точок* $x^i = (x_1^i, x_2^i, \dots, x_n^i)$ n -вимірного простору R^n , $i = 1, 2, \dots, r$, називається точка x , що являє собою лінійну комбінацію точок x^i , тобто $x = \sum_{i=1}^r \lambda_i x^i$ за умови, що $\lambda_i \geq 0$, $i = 1, 2, \dots, r$, і $\sum_{i=1}^r \lambda_i = 1$.

Означення 1.4. Множина D n -вимірного простору R^n називається *опуклою*, якщо опукла комбінація довільної (скінченної) кількості точок цієї множини належить цій самій множині.

Зокрема, для двох точок, тобто в разі $r = 2$, точка x описує відрізок, що з'єднує точки x^1 та x^2 й визначається співвідношенням $x = \lambda x^1 + (1 - \lambda)x^2$, де $0 \leq \lambda \leq 1$.

Означення 1.5. *Кутовими точками* опуклої множини називаються такі, які не можна подати опуклою комбінацією будь-яких двох різних точок цієї множини.

Інакше кажучи, кутова точка не може бути внутрішньою точкою жодного відрізка, що сполучає дві довільні точки опуклої множини.

Означення 1.6. *Півпростором* у просторі R^n називається множина точок $x = (x_1, x_2, \dots, x_n)$, для яких виконується співвідношення $\sum_{i=1}^n a_i x_i \leq b$, де a_i, b — сталі.

Означення 1.7. **Гіперплощиною** в n -вимірному просторі R^n називається множина точок $\mathbf{x} = (x_1, x_2, \dots, x_n)$, для яких виконується рівність $\sum_{i=1}^n a_i x_i = b$, де a_i, b — сталі.

Зауваження. Будь-яка гіперплощина являє собою перетин півпросторів $\sum_{i=1}^n a_i x_i \leq b$ і $\sum_{i=1}^n a_i x_i \geq b$.

Означення 1.8. Перетин скінченної кількості півпросторів називається **багатогранною множиною**, чи **багатогранником**.

Означення 1.9. Система обмежень

$$\sum_{i=1}^n a_{ij} x_i \leq b_j, \quad j = 1, 2, \dots, m,$$

називається **сумісною**, якщо багатогранна множина, задана цією системою, містить хоча б одну точку, тобто не порожня.

У лінійній алгебрі доведено, що півпростір являє собою опуклу множину й перетин двох або більше опуклих множин — також опукла множина. На підставі цих результатів доведено, що множина допустимих розв'язків ЗЛП — багатогранна множина.

Кутові точки багатогранника розв'язків ЗЛП — це вершини допустимої множини.

Теорема 1.2. Якщо цільова функція набуває максимального (мінімального) значення в якійсь точці допустимої множини D , то вона досягає його в кутовій точці множини D . Якщо цільова функція досягає максимуму (мінімуму) більше ніж в одній точці, то вона набуває цього самого значення в їх опуклій комбінації.

Теорема 1.3. Якщо множина допустимих розв'язків ЗЛП обмежена, то така задача має розв'язок, який досягається в одній із вершин допустимої множини. Якщо розв'язок досягається на множині вершин допустимої множини, то будь-яка опукла комбінація цих вершин — також розв'язок ЗЛП.

Це означає, що розв'язок ЗЛП слід шукати лише в крайніх точках допустимої множини розв'язків.

Означення 1.10. Допустимий розв'язок ЗЛП називається **базисним (опорним)**, якщо система векторів обмежень A_j , що відповідає додатним компонентам вектора розв'язку $\mathbf{x}^T = (x_1, x_2, \dots, x_n)$, лінійно незалежна.

Означення 1.11. Опорний (базисний) план ЗЛП називають **невиродженим**, або **неособливим**, якщо він має не менше додатних компонент, ніж обмежень ЗЛП.

Теорема 1.4. Допустимий розв'язок $x^T = (x_1, x_2, \dots, x_n)$ ЗЛП є вершиною допустимої множини тоді й тільки тоді, коли він базисний.

Ця теорема узагальнює результат, отриманий у ході геометричного розв'язання ЗЛП на площині, тобто для розмірності $n = 2$.

Наслідок. Кожній кутовій точці множини розв'язків ЗЛП відповідають m лінійно незалежних векторів із даної системи A_1, A_2, \dots, A_n .

Підсумовуючи всі результати, доходимо таких висновків:

1) існує така точка опуклого багатогранника розв'язків ЗЛП, у якій цільова функція набуває екстремального (максимального чи мінімального) значення;

2) між опорними планами задачі та кутовими точками множини D існує взаємно однозначна відповідність;

3) із кожною кутовою точкою пов'язані m лінійно незалежних векторів повної системи n векторів.

Перебираючи всі можливі системи лінійно незалежних векторів, можна переглянути всі вершини багатогранника допустимих розв'язків і, порівнюючи між собою значення цільової функції в цих вершинах, вибрати потрібне (найбільше чи найменше). Якщо багатогранник допустимих розв'язків має забагато вершин, то їх перебір стає клопітковою задачею. У цьому разі пошук потрібної вершини має бути спрямованим. Саме ці ідеї реалізує симплексний метод — метод послідовного поліпшення плану. Його назва походить від слова “симплекс” — найпростіший багатогранник.

За допомогою цього методу можна знайти кутову точку та дослідити її на оптимальність. Якщо точка виявиться неоптимальною, то метод дає змогу знайти наступну кутову точку, у якій значення цільової функції не погіршується порівняно зі значенням у попередній точці чи наближається до оптимального. Така властивість називається *монотонністю методу*, а перехід від однієї точки до іншої — *кроком*, або *ітерацією*. Оскільки будь-який багатогранник має скінченну кількість вершин, то через скінченну кількість кроків буде знайдено точку екстремуму чи виявиться, що цільова функція обмежена на множині допустимих розв'язків, тобто не має оптимального плану.

$$\Delta_j = (\bar{c}, A_j) - c_j = z_j - c_j,$$

де $z_j = (\bar{c}, A_j)$; c_j — коефіцієнти цільової функції при відповідних небазисних змінних.

Доведено такі теореми.

Теорема 1.5. Якщо для якогось фіксованого вектора A_j виконується умова $\Delta_j = z_j - c_j < 0$, то розв'язок ЗЛП $x = (x_1, x_2, \dots, x_n)$, що відповідає даній кутовій точці, не оптимальний для задачі максимізації, і можна побудувати таку множину планів, що для кожного з них значення цільової функції більше, ніж у даній кутовій точці.

Аналогічна теорема правдива й для задачі мінімізації.

Теорема 1.6. Якщо для якогось фіксованого вектора A_j виконується умова $\Delta_j = z_j - c_j > 0$, то розв'язок ЗЛП $x = (x_1, x_2, \dots, x_n)$, що відповідає даній кутовій точці, не оптимальний для задачі мінімізації, і можна побудувати таку множину планів, що для кожного з них значення цільової функції менше, ніж у даній кутовій точці.

Умову оптимальності розв'язку можна формулювати так.

Теорема 1.7. Якщо для якогось опорного плану ЗЛП $x = (x_1, x_2, \dots, x_n)$ усі симплекс-різниці Δ_j недодатні ($\Delta_j \leq 0$), то x — оптимальний розв'язок задачі мінімізації. Якщо ж усі симплекс-різниці невід'ємні ($\Delta_j \geq 0$) для якогось опорного плану $x = (x_1, x_2, \dots, x_n)$, то x — оптимальний розв'язок задачі максимізації.

Отже, значення симплекс-різниць Δ_j — це оцінки плану, і саме за ними визначають порядок введення до базису нових векторів.

Вектор, який вводять до базису, називається *провідним*, або *розв'язувальним*. Однак введення кожного нового вектора до базису потребує виведення якогось із тих, що утворюють поточний базис. Це означає, що для базисних векторів також має бути ознака, за якою можна визначити вектор, котрий потрібно вивести з базису, без шкоди для подальшого пошуку остаточного розв'язку. Цей вектор визначають за індексом додатної координати провідного вектора, що утворює найменше відношення з відповідною координатою вектора $A_0 = b$. Як саме обчислити це відношення, буде показано в алгоритмі методу.

Якщо на певному етапі пошуку оптимального розв'язку ЗЛП виявиться, що для якогось небазисного вектора A_k не виконується умова оптимальності, однак цей вектор не містить жодної додатної координати, то це означає, що відповідна цільова функція необмежена на множині допустимих розв'язків і ЗЛП не має оптимального розв'язку.

Щоб розв'язати задачу чи переконатися в тому, що вона не має розв'язку, потрібно виконати обчислення в зазначеному порядку.

1. Розкласти за початковим базисом усі небазисні вектори, тобто знайти всі коефіцієнти x_{ij} , $k = 1, m; j = 1, n$, за формулою

$$(x_{1j}, x_{2j}, \dots, x_{mj})^T = B^{-1}(a_{1j}, a_{2j}, \dots, a_{mj})^T$$

де B — базисна матриця.

Для вектора A_0 розклад відомий: його компоненти збігаються з нульовими компонентами заданого опорного розв'язку, тобто $x_{j0} = x_s^{(j)}$, де s — номери m додатних координат вихідного опорного розв'язку.

2. Для кожного j ($j = 1, n$) обчислити оцінку — симплекс-різницю

$$\Delta_j = z_j - c_j = \sum c_{s_k} x_{kj} - c_j,$$

де c_j — коефіцієнти при невідомих змінних x_j у цільовій функції; c_{s_k} — коефіцієнти цільової функції при базисних змінних (усього m); x_{kj} — компоненти відповідного вектора A_j . Якщо всі коефіцієнти c_{s_k} об'єднати в m -вимірний вектор \bar{c} , а коефіцієнти x_{kj} позначити як вектор-стовпець A_j , то симплекс-різницю Δ_j можна записати через скалярний добуток векторів у вигляді $\Delta_j = (\bar{c}, A_j) - c_j$.

Якщо всі $\Delta_j \geq 0$, то процес розв'язання задачі максимізації завершено. Розглянутий опорний розв'язок оптимальний, а оптимум цільової функції можна визначити зі співвідношення $z_{\max} = (\bar{c}, A_j)$.

Якщо є $\Delta_j < 0$, то переходимо до наступного кроку.

3. З'ясувати, чи існує хоча б одна така від'ємна оцінка Δ_j , що координати x_{ij} вектора A_j в заданому базисі від'ємні. Якщо така оцінка є, то процес розв'язання задачі завершено, і цільова функція задачі не обмежена (зверху) на допустимій множині, а не то переходимо до наступного кроку.

4. Вибрати найменшу оцінку з $\Delta_j < 0$ (нехай це буде Δ_r); обчислити відношення x_{k0} / x_{kr} для всіх k , для яких $x_{kr} > 0$, і знайти мінімальне з цих відношень (нехай це буде $\theta_s = x_{s0} / x_{sr}$).

5. Перейти до нового опорного розв'язку, базис якого утворюється заміною вектора A_r у попередньому базисі вектором A_s . Координати всіх векторів A_0, A_1, \dots, A_n у новому базисі обчислюють за такими формулами:

$$x'_{ij} = \theta_j = \frac{x_{rj}}{x_{rs}}; x_{kj} = x_{kj} - \theta_j x_{ks}, \quad j = \overline{0, n}, \quad k = \overline{1, m}, \quad k \neq r.$$

6. Перейти до виконання пункту 2 та, якщо потрібно, виконати подальші обчислення відносно нового базису.

Зазначена послідовність дій утворює *алгоритм симплекс-методу*.

Нагадаємо, що кожен перехід до нового базису (а отже, до нової вершини допустимої множини розв'язків ЗЛП) називається *кроком* або, частіше, *ітерацією симплекс-методу*. Доведено, що за скінченну кількість кроків (ітерацій) процес обчислень закінчиться або на п. 2, коли знайдений розв'язок оптимальний, або на п. 3, коли буде виявлено, що оптимального розв'язку не існує.

Пошук початкового опорного розв'язку

Для виконання п. 1 алгоритму симплекс-методу потрібно знати якийсь опорний розв'язок і матрицю B^{-1} , обернену до базисної матриці B . Якщо у вихідній задачі досить багато обмежень, то для визначення початкового базису, тобто перевірки лінійної незалежності векторів, потрібно обчислювати визначники матриць із досить великою розмірністю, що разом із визначенням оберненої матриці B^{-1} потребує значного обсягу обчислень. Тому на практиці зазвичай зручніше починати обчислення з опорного розв'язку, якому відповідає одиничний базис, тобто $B = E$. Якщо такого опорного розв'язку в задачі немає, то за певними правилами її зводять до іншого вигляду, що має очевидний опорний розв'язок з одиничним базисом — так званої *канонічної форми ЗЛП*.

Означення 1.12. Система обмежень ЗЛП називається записаною в *канонічній формі*, якщо вона складається лише з рівнянь, що мають додатні вільні члени, і кожне з цих рівнянь містить одну змінну з коефіцієнтом 1, яка не входить до інших рівнянь (тобто в інших рівняннях коефіцієнт при ній дорівнює 0).

Це означає, що обмеження ЗЛП в канонічній формі мають вигляд

$$\left\{ \begin{array}{l} x_1 + a_{1,m+1}x_{m+1} + \dots + a_{1n}x_n = b_1, \\ x_2 + a_{2,m+1}x_{m+1} + \dots + a_{2n}x_n = b_2, \\ \dots \\ x_m + a_{m,m+1}x_{m+1} + \dots + a_{mn}x_n = b_m. \end{array} \right. \quad (1.11)$$

Розширена задача має одиничний базис $B = (A_{n+1}, A_{n+2}, \dots, A_{n+m})$, який називають *штучним*. Ідея методу штучного базису полягає в тому, щоб послідовним виведенням із базису штучних векторів одержати базис початкової задачі та водночас виявити, чи сумісна система обмежень початкової задачі, тобто чи має задача хоча б один розв'язок.

Доведено такі твердження.

Теорема 1.8. Якщо існує принаймні один план початкової ЗЛП, то штучні змінні розширеної задачі дорівнюють нулю: $x_{n+1} = x_{n+2} = \dots = x_{n+m} = 0$.

Теорема 1.9. Якщо оптимальний розв'язок розширеної ЗЛП не містить жодної додатної штучної змінної $x_0 = (x_1, x_2, \dots, x, 0, \dots, 0)$, а складається лише з t додатних і $n - t$ нульових змінних початкової задачі, то план $x_0 = (x_1, x_2, \dots, x_n)$ — оптимальний розв'язок початкової ЗЛП.

Для розв'язання розширеної ЗЛП застосовують звичайний симплекс-метод.

Зауваження. Якщо до зведення до стандартної форми задача мала обмеження типу “ \leq ”, то додаткові змінні, що перетворюють ці нерівності в рівняння, використовують як базисні, тобто в цьому разі кількість штучних змінних можна зменшити. Нагадаємо, що штучні змінні відрізняються від додаткових тим, що вони входять у цільову функцію не з нульовим, а з досить великим коефіцієнтом M ($M \gg 0$ для задачі мінімізації й $M \ll 0$ для максимізації). Для обмежень типу “ \geq ” для переходу до канонічної форми потрібні як додаткові, так і штучні змінні. Додаткова змінна, що дає змогу перетворити нерівність у строгу рівність, у цьому разі має знак “ $-$ ”, тобто відповідна їй компонента опорного плану від'ємна, що неприпустимо в такій постановці задачі. Тоді штучна змінна допомагає уникнути подібної ситуації.

У разі ручного розв'язання ЗЛП зручно оформляти його результати у вигляді таблиць, кожна з яких називають *повною симплекс-таблицею*, що відповідає даному опорному розв'язку. У таблиці записують матрицю обмежень $A = (A_1, A_2, \dots, A_n)$ і вектор вільних членів $A = b$. У її заголовку разом з іменами векторів A_1, A_2, \dots, A_n записують коефіцієнти цільової функції при відповідних змінних. У першому стовпці записують базисні вектори, у другому — коефіцієнти цільової функції при відповідних базисних змінних (вектор \bar{c}),

в останньому — оцінки базисних векторів, за якими визначають, котрий із них буде виведено з базису. Останній рядок таблиці містить оцінки Δ_j , за якими визначають оптимальність поточного опорного розв'язку.

Оцінка Δ_0 дорівнює значенню цільової функції для даного опорного розв'язку:

$$\Delta_0 = z = (\bar{c}, x) = c_{i1}x_{i1} + c_{i2}x_{i2} + \dots + c_{im}x_{im}.$$

Розглянемо застосування симплекс-таблиць до розв'язання такої ЗЛП.

Приклад 1.6. Знайти максимум функції $2x_1 - x_2 + 3x_3 + 2x_4$ для обмежень

$$\begin{cases} 2x_1 + 3x_2 - x_3 + 2x_4 + x_5 & = 4, \\ x_1 - 2x_2 + 5x_3 - 3x_4 - x_6 & = 1, \\ 4x_1 + 10x_2 + 3x_3 + x_4 + x_7 & = 8, \\ x_j \geq 0, j = 1, 2, \dots, 7, \end{cases}$$

якщо відомий опорний розв'язок цієї задачі $(1; 0; 0; 0; 2; 0; 4)$.

Матриця обмежень задачі має вигляд

$$A = (A_1, A_2, A_3, A_4, A_5, A_6, A_7) = \begin{pmatrix} 2 & 3 & -1 & 2 & 1 & 0 & 0 \\ 1 & -2 & 5 & -3 & 0 & -1 & 0 \\ 4 & 10 & 3 & 1 & 0 & 0 & 1 \end{pmatrix}.$$

Легко перевірити, що її ранг дорівнює 3, тобто будь-які три її вектори-стовпці лінійно незалежні. Заданий опорний розв'язок не вироджений, бо має три додатні координати. Його базис утворюють вектори A_1, A_5, A_7 , тобто

$$B = (A_1, A_5, A_7) = \begin{pmatrix} 2 & 1 & 0 \\ 1 & 0 & 0 \\ 4 & 0 & 1 \end{pmatrix};$$

$$B^{-1} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -2 & 0 \\ 0 & -4 & 1 \end{pmatrix}.$$

Можна розкласти водночас усі небазисні вектори A_2, A_3, A_4, A_6 за цим базисом:

$$\begin{pmatrix} x_{12} & x_{13} & x_{14} & x_{16} \\ x_{22} & x_{23} & x_{24} & x_{26} \\ x_{32} & x_{33} & x_{34} & x_{36} \end{pmatrix} = B^{-1}(A_2, A_3, A_4, A_6) = \\ = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -2 & 0 \\ 0 & -4 & 1 \end{pmatrix} \begin{pmatrix} 3 & -1 & 2 & 0 \\ -2 & 5 & -3 & -1 \\ 10 & 3 & 1 & 0 \end{pmatrix} = \begin{pmatrix} -2 & 5 & -3 & -1 \\ 7 & -11 & 8 & 2 \\ 18 & -17 & 13 & 4 \end{pmatrix}.$$

Координати небазисних векторів переобчислювати не потрібно, якщо в друге рівняння-обмеження ввести штучну змінну $x_8 \geq 0$. Тоді матриця A збільшиться на один стовпець A_8 , але базисна матриця $B = (A_5, A_7, A_8)$ буде одиничною, отже й обернена до неї — також одиничною, тобто $B^{-1} = E$, і коефіцієнти x_{ij} небазисних векторів залишаться початковими.

Змінивши порядок запису обмежень, отримаємо

$$A_0 = \begin{pmatrix} 4 \\ 8 \\ 1 \end{pmatrix}; A = \begin{pmatrix} 2 & 3 & -1 & 2 & 1 & 0 & 0 & 0 \\ 4 & 10 & 3 & 1 & 0 & 0 & 1 & 0 \\ 1 & -2 & 5 & -3 & 0 & -1 & 0 & 1 \end{pmatrix};$$

$$B = (A_5, A_7, A_8) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

У цьому разі опорний розв'язок можна задати через координати вектора A_0 у вигляді $\mathbf{x}^{(1)} = (0; 0; 0; 0; 4; 0; 8; 1)$.

Заповнимо першу симплекс-таблицю, що відповідає опорному розв'язку $x^{(1)}$. Будемо вважати, що коефіцієнт $c_0 = 0$.

Базис	$c_{\text{баз}}$	A_0	2	-1	3	2	0	0	0	$-M$	θ
			A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	
A_3	0	4	2	3	-1	2	1	0	0	0	$\frac{4}{-1}$
A_7	0	8	4	10	3	1	0	0	1	0	$\frac{8}{3}$
A_8	$-M$	1	1	-2	5	-3	0	-1	0	1	$\frac{1}{5}$
	Δ_j	$-M$	$-M-2$	$2M+1$	$-5M-3$	$3M-2$	0	M	0	0	

Нагадаємо, що симплекс-різниці обчислюють за формулою

$$\Delta_j = z_j - c_j = \sum_{k=1}^m c_{ik} x_{k0} - c_j.$$

Тому

$$\Delta_0 = z = (c_{\text{баз}}, A_0) = 0 \cdot 4 + 0 \cdot 8 + (-M) \cdot 1 = -M,$$

$$\Delta_1 = z_1 - c_1 = (c_{\text{баз}}, A_1) - c_1 = 0 \cdot 2 + 0 \cdot 4 + (-M) \cdot 1 - 2 = -M - 2,$$

$$\Delta_2 = z_2 - c_2 = (c_{\text{баз}}, A_2) - c_2 = 0 \cdot 3 + 0 \cdot 10 + (-M) \cdot (-2) + 1 = 2M + 1,$$

$$\Delta_3 = z_3 - c_3 = (c_{\text{баз}}, A_3) - c_3 = 0 \cdot (-1) + 0 \cdot 3 + (-M) \cdot 5 - 3 = -5M - 3,$$

$$\Delta_4 = z_4 - c_4 = (c_{\text{баз}}, A_4) - c_4 = 0 \cdot 2 + 0 \cdot 1 + (-M) \cdot (-3) - 2 = 3M - 2,$$

$$\Delta_6 = z_6 - c_6 = (c_{\text{баз}}, A_6) - c_6 = 0 \cdot 0 + 0 \cdot 0 + (-M) \cdot (-1) - 0 = M.$$

Оскільки $M \gg 0$, то симплекс-різниці Δ_1 і Δ_3 від'ємні, тобто критерій оптимальності для цього розв'язку не виконується. Однак вектори A_1, A_3 мають додатні координати, тому можна шукати розв'язок, якому відповідає більше значення цільової функції, тобто план задачі можна поліпшити. Позаяк $\Delta_3 < \Delta_1$, то як провідний (розв'язувальний) беруть вектор A_3 та вводять його до базису.

Щоб визначити, який вектор потрібно вивести з базису, заповнюють останній стовпець симплекс-таблиці, а саме обчислюють відно-

шення координат вектора A_0 до відповідних координат провідного вектора A_3 , тобто $\bar{\theta} = (-4; 8/3; 1/5)$. Оскільки від'ємні координати цього вектора не беруть до уваги, а мінімальний елемент вектора $\bar{\theta}$ визначають серед його додатних компонент ($\min \theta = \min(8/3; 1/5) = 1/5$), то це означає, що з базису слід вивести третій вектор стовпця “Базис”, тобто A_8 . Рядок, що відповідає цьому вектору, називають провідним, а елемент, що стоїть на перетині провідного рядка та провідного стовпця, також називається провідним.

Зауваження. Штучну змінну, виведену з базису, можна надалі не розглядати, а симплекс-таблиця при цьому скорочується на один стовпець.

Елементи матриці $A = (A_0, A_1, \dots, A_7)$ переобчислюють так, щоб на місці нововведеного стовпця A_3 стояв одиничний вектор з одиницею на третій позиції ($r = 3$). Для цього всі елементи провідного рядка ділять на провідний елемент x_{33} та на його місці одержують 1.

Інші елементи в провідному вектор-стовпці виключають за допомогою лінійних комбінацій перетвореного провідного рядка з усіма іншими рядками матриці A , тобто переходять до нового опорного розв'язку методом виключення Жордана — Гаусса.

Наприклад, щоб виключити елемент $x_{13} = -1$, перший рядок складають із третім (провідним) без додаткового множення, тому що $-1 + 1 = 0$, а щоб виключити елемент $x_{23} = 3$, провідний рядок попередньо множать на мінус 3.

Загалом цей процес можна подати *основними формулами методу Гаусса*:

$$x'_{rj} = \theta_j = \frac{x_{rj}}{x_{rs}}, \quad k \neq r, \quad x'_{kj} = x_{kj} = \theta_j x_{ks}.$$

Для обчислення симплекс-різниць також можна використати подібну формулу, а саме:

$$\Delta'_j = \Delta_j - \frac{x_{rj}}{x_{rs}} \Delta_s.$$

Після обчислень отримуємо послідовність таблиць, в останній з яких або всі симплекс-різници невід'ємні, тобто знайдено оптимальний розв'язок, або провідний вектор не має додатних компонент,

тобто цільова функція необмежена на множині допустимих розв'язків, і ЗЛП не має оптимального плану.

В одержаній симплекс-таблиці симплекс-різниці $\Delta_1, \Delta_2, \Delta_4$, та Δ_6 від'ємні, тому розв'язок $x^{(2)} = (0; 0; 1/5; 0; 21/5; 0; 37/5)$ не оптимальний.

Базис	$e_{\text{баз}}$	A_0	2	-1	3	2	0	0	0	θ
			A_1	A_2	A_3	A_4	A_5	A_6	A_7	
A_5	0	$\frac{21}{5}$	$\frac{11}{5}$	$\frac{13}{5}$	0	$\frac{7}{5}$	1	$-\frac{1}{5}$	0	3
A_7	0	$\frac{37}{5}$	$\frac{17}{5}$	$\frac{56}{5}$	0	$\frac{14}{5}$	0	$\frac{3}{5}$	1	$\frac{37}{14}$
A_3	3	$\frac{1}{5}$	$\frac{1}{5}$	$-\frac{2}{5}$	1	$-\frac{3}{5}$	0	$-\frac{1}{5}$	0	-3
		$\frac{3}{5}$	$-\frac{7}{5}$	$-\frac{1}{5}$	0	$-\frac{19}{5}$	0	$-\frac{3}{5}$	0	

Оскільки різниця $\Delta_4 = -19/5$ найменша, то як провідний візьмемо вектор A_4 та введемо його до базису. Мінімальний додатний елемент вектора $\bar{\theta} = (3; 37/14; -3)$ дорівнює $37/14$, отже з базису слід вилючити другий його вектор, тобто A_7 . Переобчислимо елементи матриці $A = (A_0, A_1, \dots, A_7)$ так, щоб на місці провідного елемента x_{42} була одиниця, а інші елементи провідного стовпця дорівнювали нулю. Отримаємо таку симплекс-таблицю:

Базис	$e_{\text{баз}}$	A_0	2	-1	3	2	0	0	0	θ
			A_1	A_2	A_3	A_4	A_5	A_6	A_7	
A_5	0	$\frac{1}{5}$	$\frac{1}{2}$	-3	0	0	1	$-\frac{1}{2}$	$-\frac{1}{2}$	
A_4	2	$\frac{37}{14}$	$\frac{17}{14}$	4	0	1	0	$\frac{3}{14}$	$\frac{5}{14}$	
A_3	3	$\frac{25}{14}$	$\frac{13}{14}$	2	1	0	0	$-\frac{1}{14}$	$\frac{5}{14}$	
	Δ_j	$\frac{149}{14}$	$\frac{45}{14}$	15	0	0	0	$\frac{3}{14}$	$\frac{25}{14}$	

У ній усі симплекс-різниці невід'ємні, тому останній опорний план $\mathbf{x}^{(3)} = (0; 0; 25/14; 37/14; 1/2; 0; 0)$ оптимальний. Для нього цільова функція набуває значення 149/14.

Модифікований симплекс-метод

У разі розв'язання ЗЛП на ЕОМ (а це зазвичай задачі великої розмірності) істотна кількість обчислень на кожній ітерації, бо від цього залежить, скільки потрібно пам'яті й часу та яка має бути точність обчислень.

Аналіз симплекс-методу показав, що для визначення оцінок Δ_j не потрібно переобчислювати матрицю $A = (A_0, A_1, \dots, A_n)$. Виявлено, що $\Delta_j^{(l)}$ (l — номер ітерації) можна визначити так:

$$\Delta_j^{(l)} = (\mathbf{c}_{\text{баз}}, A_j^{(l)}) - c_j = (\mathbf{c}_{\text{баз}}, B^{-1}(l)A_j) - c_j = (\mathbf{u}^{(l)}, A_j) - c_j,$$

де $\mathbf{c}_{\text{баз}} = (c_{l1}, \dots, c_{lm})$ — коефіцієнти цільової функції при базисних змінних; $\mathbf{u}^{(l)} = (u_{l1}, \dots, u_{lm}) = \mathbf{c}_{\text{баз}} B^{-1}(l)$. Координати u_{li} вектора $\mathbf{u}^{(l)}$ називаються симплекс-множниками.

Крім того, існують прості формули переходу від $B^{-1}(l)$ до $B^{-1}(l+1)$, тобто від оберненої до базисної матриці l -ї ітерації до матриці $B^{-1}(l+1)$, оберненої до базисної матриці $(l+1)$ -ї ітерації. Це відомі *формули прямокутника* (метод Жордана — Гаусса):

$$b_{kj}^{(l+1)} = \begin{cases} b_{kj}^{(l)} - \frac{x_{rj}^{(l)}}{x_{rs}^{(l)}} b_{ks}^{(l)}, & r \neq j, \\ \frac{b_{rj}^{(l)}}{x_{rs}^{(l)}}, & r = k. \end{cases} \quad (1.12)$$

Алгоритм модифікованого симплекс-методу має такий вигляд.

1. Обчислити симплекс-різницю $\Delta_j = \mathbf{u} A_j - c_j$, де $\mathbf{u} = \mathbf{c}_{\text{баз}} B^{-1}$ — вектор симплекс-множників; B^{-1} — матриця, обернена до базисної. Якщо $\Delta_j \geq 0$, $j = \overline{1, n}$, то обчислення закінчено, базисний розв'язок оптимальний. Якщо існує таке j , що $\Delta_j < 0$, то перейти до наступного кроку.

2. Вибрати таке k , що $\Delta_k < 0$ (зазвичай найменше). Обчислимо нові вектори $\alpha = B^{-1}A_s$ і $\beta = B^{-1}b$. Якщо вектор α_k не має додатних компонент, тобто $\alpha_k \leq 0$, то обчислення закінчено; цільова функція не обмежена зверху на допустимій множині. Якщо існують такі i , що $\alpha_{ik} > 0$, то перейти до наступного кроку.

3. Обчислити $\theta = \beta_i / \alpha_{ik}$ для всіх $\alpha_{ik} > 0$ та вибрати серед них мінімальне. Нехай це θ , що стоїть на r -му місці. Вектор A_s увести в базис, а вектор A_r вивести з нього. Матрицю $B^{(-1)}(l)$ на l -й ітерації обчислити за матрицею $B^{(-1)}(l-1)$ (отриманою на попередній ітерації) за формулами (1.12). Для канонічної ЗЛП $B^{-1}(0) = E$ — одинична матриця.

Теорія двоїстості та двоїсті оцінки лінійних оптимізаційних моделей

Нехай якесь підприємство виробляє n видів продукції, вартість одиниці кожного з них становить c_j ($j = 1, 2, \dots, n$). При цьому підприємство витрачає m видів ресурсів у кількостях b_i ($i = 1, 2, \dots, m$), a_{ij} — технологічна норма використання i -го ресурсу для виробництва одиниці j -го продукту. Задача виробничого планування полягає в тому, щоб визначити, у яких кількостях x_j ($j = 1, 2, \dots, n$) слід виробляти кожен із продуктів, щоб прибуток підприємства був максимальним.

Очевидно, що внесок кожного з використовуваних ресурсів у сумарний дохід підприємства різний. Оцінивши цей внесок, можна шукати способи оптимального використання ресурсів і тим самим збільшувати дохід.

Позначимо як y_i ($i = 1, 2, \dots, m$) вартість одиниці i -го ресурсу. Тоді вартість ресурсів, витрачених на виготовлення одиниці j -ї продукції ($j = 1, 2, \dots, n$), дорівнює

$$a_{1j}y_1 + a_{2j}y_2 + \dots + a_{mj}y_m$$

Для кожного виду продукції виконується нерівність

$$a_{1j}y_1 + a_{2j}y_2 + \dots + a_{mj}y_m \geq c_j \\ (j = 1, 2, \dots, n).$$

$$A_j = \begin{pmatrix} a_{1j} \\ a_{2j} \\ \dots \\ a_{mj} \\ c_j \end{pmatrix},$$

де a_{ij} — кількість i -го ресурсу, витраченого в разі одиничної інтенсивності технології; c_j — кількість одержаної при цьому продукції.

Передбачено, що всі ресурси є в заданих кількостях b_i , $i = \overline{1, m}$. Задача полягає в тому, щоб визначити інтенсивності, з якими потрібно використовувати різні технології, щоб сумарна кількість готової продукції була максимальною.

Запишемо математичну модель цієї задачі. Нехай x_1, x_2, \dots, x_n — шукані інтенсивності. Будемо вважати, що витрати ресурсів і обсяг випуску готової продукції пропорційні відповідній інтенсивності. Тоді загальна кількість продукції виражається сумою $c_1x_1 + c_2x_2 + \dots + c_nx_n$, а використаний у різних технологіях i -й ресурс не перевищує наявних запасів b_i , тобто для всіх $i = 1, 2, \dots, m$ виконується нерівність $a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n \leq b_i$. Усі x_j відповідно до їх змісту невід'ємні, тобто $x_j \geq 0$, $j = \overline{1, n}$. Отже, маємо таку ЗЛП:

$$c_1x_1 + c_2x_2 + \dots + c_nx_n \rightarrow \max,$$

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \leq b_1, \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n \leq b_2, \\ \dots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \leq b_m, \\ x_j \geq 0, \quad j = 1, 2, \dots, n. \end{cases}$$

Щоб об'єктивно порівняти загальні витрати з результатом, потрібно визначити частку кожного з використовуваних у виробництві ресурсів. Нехай y_i ($i = 1, 2, \dots, m$) — частка i -го ресурсу. Тоді функція $b_1y_1 + b_2y_2 + \dots + b_my_m$ визначає сумарну оцінку всіх ресурсів, які має підприємство. У свою чергу, для часток ресурсів виконуються нерівності

$$a_{1j}y_1 + a_{2j}y_2 + \dots + a_{mj}y_m \geq c_j \quad (j = 1, 2, \dots, n),$$

оскільки реальні витрати підприємства на виробництво кожного продукту не можуть бути меншими, ніж його остаточна вартість, бо більша вартість не виробляється з “нічого”.

Розв'язання задачі мінімізації функції

$$b_1y_1 + b_2y_2 + \dots + b_my_m \rightarrow \min$$

для заданих обмежень

$$\begin{cases} a_{11}y_1 + a_{21}y_2 + \dots + a_{m1}y_m \geq c_1, \\ a_{12}y_1 + a_{22}y_2 + \dots + a_{m2}y_m \geq c_2, \\ \dots\dots\dots \\ a_{1n}y_1 + a_{2n}y_2 + \dots + a_{mn}y_m \geq c_n \end{cases}$$

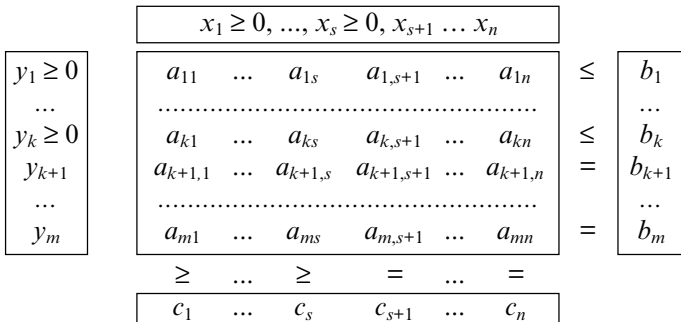
і прямих обмежень на змінні

$$y_1 \geq 0, y_2 \geq 0, \dots, y_m \geq 0$$

дає змогу порівняти витрати підприємства з кінцевим результатом для різних технологічних способів виробництва.

У цьому разі першу задачу інтерпретують як задачу виробничого планування з одним продуктом, m виробничими факторами й n технологіями, а двоїсту до неї — як задачу пошуку частки кожного з ресурсів, що витрачені на одиницю продукції, виробленої різними технологічними способами.

Що стосується загальної теорії двоїстості в ЛП, то існують правила переходу від прямої ЗЛП до двоїстої та навпаки. Подамо їх у вигляді схеми:



Як видно з наведеної схеми, переходячи від прямої задачі до двоїстої, потрібно виконати такі дії:

- 1) змінити тип екстремуму на протилежний (максимум на мінімум або навпаки);
- 2) поміняти місцями вектор коефіцієнтів цільової функції $c = (c_1, c_2, \dots, c_n)$ і стовпець обмежень $b = (b_1, b_2, \dots, b_m)$;
- 3) транспонувати матрицю обмежень A ;
- 4) індексам змінних, які задовольняють прямим обмеженням, постановити у відповідність номери обмежень двоїстої задачі, що виконуються як нерівності (якщо прямі обмеження на змінні не накладено, то відповідні обмеження двоїстої задачі виконуються як рівняння);
- 5) і навпаки, номерам обмежень прямої задачі, які виконуються як нерівності, постановити у відповідність індекси двоїстих змінних, що задовольняють прямим обмеженням; обмеженням-рівнянням відповідають двоїсті змінні, що не мають обмежень.

Згідно з наведеними правилами можна сформулювати важливу особливість ЗЛП — *симетричність* відношення двоїстості, яка полягає в тому, що задача, двоїста до двоїстої, збігається з прямою ЗЛП. Тому завжди говорять про пару взаємно двоїстих задач.

Розглянемо процес побудови двоїстої задачі на прикладі.

Приклад 1.7. Нехай пряма ЗЛП — це задача максимізації, обмеження якої складаються з нерівностей і рівнянь, а прямі обмеження накладено лише на частину змінних:

$$3x_1 + 2x_2 - x_3 \rightarrow \max,$$
$$\begin{cases} 2x_1 - 2x_2 + x_3 - 3x_4 \leq 8, \\ x_1 + 4x_2 - 2x_3 + x_4 = 4, \\ 2x_1 - x_2 - x_3 \leq 2, \\ x_2 \geq 0, \quad x_4 \geq 0. \end{cases}$$

Двоїста до неї задача являє собою задачу мінімізації лінійної функції, коефіцієнти якої — обмеження прямої задачі, тобто вектор $b = (8; 4; 2)$. Двоїста задача має чотири обмеження (за кількістю змінних початкової задачі). Обмеження двоїстої задачі також складаються з рівнянь і нерівностей, а саме: перше й третє — рівняння, оскільки відповідні

$$b_1 y_1 + b_2 y_2 + \dots + b_m y_m \rightarrow \min,$$

$$\begin{cases} a_{11}y_1 + a_{12}y_2 + \dots + a_{m1}y_m \geq c_1, \\ a_{21}y_1 + a_{22}y_2 + \dots + a_{m2}y_m \geq c_2, \\ \dots \\ a_{1n}y_1 + a_{2n}y_2 + \dots + a_{mn}y_m \geq c_n, \\ y_j \geq 0, \quad j = 1, 2, \dots, m. \end{cases} \quad (1.14)$$

Задача (1.14) має таку саму симетричну форму, що й задача (1.13). Така пара задач називається *симетричною* парою взаємно двоїстих задач. У матричному вигляді їх можна записати так:

<p><i>пряма</i></p> $(c, x) \rightarrow \max,$	<p><i>двоїста</i></p> $(b, y) \rightarrow \min$
$\begin{cases} Ax \leq b, \\ x \geq 0, \end{cases}$	$\begin{cases} A^T y \geq c, \\ y \geq 0 \end{cases}$

або

<p><i>пряма</i></p> $(c, x) \rightarrow \min,$	<p><i>двоїста</i></p> $(b, y) \rightarrow \max$
$\begin{cases} Ax \geq b, \\ x \geq 0, \end{cases}$	$\begin{cases} A^T y \leq c, \\ y \geq 0. \end{cases}$

Якщо одну із задач задано в стандартній формі, тобто її система обмежень складається лише з рівнянь і всі змінні мають прямі обмеження, то система обмежень двоїстої задачі являє собою систему нерівностей, а двоїсті змінні не мають обмежень. Така пара задач називається *несиметричною* парою взаємно двоїстих задач. Матричний запис обох задач такий:

<p><i>пряма</i></p> $(c, x) \rightarrow \max,$	<p><i>двоїста</i></p> $(b, y) \rightarrow \min$
$\begin{cases} Ax = b, \\ x \geq 0, \end{cases}$	$A^T y \geq c$

або

$$\begin{array}{ll} \text{пряма} & \text{двоїста} \\ (c, x) \rightarrow \min, & (b, y) \rightarrow \max \\ \left\{ \begin{array}{l} Ax = b, \\ x \geq 0, \end{array} \right. & A^T y \geq c. \end{array}$$

Теореми двоїстості та їх застосування

Двоїсті задачі мають не лише формальний зв'язок, їх розв'язки тісно пов'язані між собою. Це дає змогу розв'язувати лише одну задачу й за отриманим розв'язком знаходити розв'язок іншої.

Виявлено, що значення цільової функції в задачі максимізації, записаній у стандартній формі, для будь-якого її допустимого розв'язку x не перевищує значення цільової функції двоїстої задачі (задачі мінімізації) для будь-якого її допустимого розв'язку y , тобто

$$(c, x) \leq (b, y).$$

Якщо для якихось векторів x^0, y^0 значення цільових функцій збігаються, тобто

$$(c, x^0) \leq (b, y^0),$$

то x^0, y^0 — оптимальні розв'язки відповідних задач.

Фундаментальні властивості двоїстих задач можна сформулювати у вигляді таких теорем.

Теорема 1.10. Якщо одна з двоїстих задач має оптимальний розв'язок, то й інша також має оптимальний розв'язок, причому їх оптимуми (значення цільової функції) збігаються. Якщо цільова функція однієї із задач не обмежена зверху (знизу) на своїй допустимій множині, то інша не має допустимих розв'язків.

У ході доведення цієї теореми було виявлено таку важливу з практичного погляду властивість двоїстих задач. Якщо матриця прямої задачі має початковий одиничний базис, тобто містить одиничну матрицю (нехай для визначеності одиничними є перші вектори A_1, A_2, \dots, A_m), то в кінцевій симплексній таблиці симплекс-різниці $\Delta_1, \Delta_2, \dots, \Delta_m$ визначають розв'язок двоїстої задачі, а саме

$$y_i = \Delta_i + c_i, \quad i = 1, 2, \dots, m.$$

Повернімося до прикладу 1.6. Початковий базис цієї задачі утворюють вектори A_5, A_7, A_8 , причому змінна x_8 штучна, і після виведення з базису її було виключено з таблиці. Тому значення двоїстої змінної потрібно визначити за значенням симплекс-різниці додаткової змінної x_6 .

Теорема 1.11 (двоїстий критерій оптимальності). Допустимі розв'язки $\mathbf{x} = (x_1, x_2, \dots, x_n)$ та $\mathbf{y} = (y_1, y_2, \dots, y_m)$ пари двоїстих задач тоді й тільки тоді оптимальні, коли для кожного $j = 1, 2, \dots, n$ виконується рівність

$$(a_{1j}y_1 + a_{2j}y_2 + \dots + a_{mj}y_m - c_j) x_j = 0, \quad j = 1, 2, \dots, n. \quad (1.15)$$

Якщо задачі симетричні, то мають виконуватися ще й рівності

$$(a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n - b_i) y_i = 0, \quad i = 1, 2, \dots, m. \quad (1.16)$$

Співвідношення (1.15), (1.16) називаються *умовами доповнювальної нежорсткості*.

Двоїстий критерій оптимальності дає змогу за розв'язком однієї із задач визначити розв'язок іншої: якщо в разі підстановки в систему обмежень прямої задачі компонент її оптимального плану i -те обмеження виконується як строга нерівність, то відповідна двоїста змінна дорівнює нулю, тобто $y_i = 0$. Якщо ж i -те обмеження виконується як рівність, то відповідна двоїста змінна додатна, тобто $y_i > 0$.

Отже, після визначення оптимального плану прямої задачі знайдений розв'язок підставляють у її систему обмежень. За знаками обмежень визначають набір додатних координат вектора двоїстих змінних $\mathbf{y} = (y_1, y_2, \dots, y_m)$. Потім переходять до обмежень двоїстої задачі та розглядають їх як систему рівнянь відносно додатних двоїстих змінних (нульові двоїсті змінні вилучають із системи). Якщо отримана так система рівнянь має єдиний розв'язок (її матриця квадратна), то це оптимальний розв'язок двоїстої задачі. Якщо система рівнянь вироджена, то для розв'язання задачі можна використати цільову функцію, скориставшись першою теоремою двоїстості, згідно з якою значення оптимумів обох задач збігаються, тобто

$$(\mathbf{c}, \mathbf{x}^0) = (\mathbf{b}, \mathbf{y}^0).$$

Якщо повернутися до початкової постановки взаємно двоїстих задач (наприклад, до загальної задачі виробничого планування з m інгредієнтами й n технологіями), то можна надати теоремам двоїстості економічного змісту. Зокрема, згідно з першою теоремою двоїстості, якщо одна із задач має оптимальний розв'язок, то й двоїста до неї задача також має оптимальний розв'язок. Це означає, що, визначивши максимально можливий дохід виробництва, можна знайти сумарну оцінку всіх інгредієнтів, причому такі оцінки обох задач збігаються. Якщо неможливо визначити максимально можливий дохід (цільова функція не обмежена зверху), то не можна знайти й придатні оцінки використовуваних інгредієнтів.

Післяоптимізаційний аналіз задачі

Властивості об'єктивно зумовлених двоїстих оцінок широко використовують для економічного аналізу результатів розв'язку ЗЛП та вироблення відповідних рекомендацій. У реальному виробничому процесі обсяги ресурсів можуть не відповідати плановим, тому в цьому разі природно досліджувати варіації запасів ресурсів, щоб оптимальний план залишився без змін. Така задача називається *проблемою стійкості оптимального плану*. Для її розв'язання замість вектора правих частин $\mathbf{b} = (b_1, b_2, \dots, b_m)$ розглядають змінений вектор — із приростами

$$\mathbf{b} + \Delta\mathbf{b} = (b_1 + \Delta_1, b_2 + \Delta_2, \dots, b_m + \Delta_m).$$

Зміна правої частини обмежень задачі зумовлює зміну значення оптимального плану $x^0(\mathbf{b} + \Delta\mathbf{b})$ та цільової функції $f = f(x^0(\mathbf{b} + \Delta\mathbf{b}))$. У ході визначення можливих варіацій вектора \mathbf{b} виявлено, що кожен двоїсту змінну можна розглядати як граничну (миттєву) оцінку внеску i -го ресурсу в загальний дохід $Z = (c, \mathbf{x})$ для оптимального розв'язку x^0 . Інакше кажучи, величина y_i^0 ($i = 1, 2, \dots, m$) дорівнює приросту доходу внаслідок збільшення i -го ресурсу на одиницю за умови, що ресурси використовують оптимально. Саме тому двоїсті змінні називаються двоїстими оцінками чи тіньовими цінами; інколи їх називають об'єктивно зумовленими оцінками.

На підставі властивостей двоїстих ЗЛП можна сформулювати такі важливі з економічного погляду наслідки.

1. Якщо для певного оптимального плану прямої задачі i -те обмеження виконується як строга нерівність, то оптимальне значення двоїстої змінної y_i дорівнює нулю, тобто якщо $a_{i1}x_1^0 + a_{i2}x_2^0 + \dots + a_{in}x_n^0 < b_i$, то $y_i = 0$. У задачі оптимального виробничого планування це означає, що якщо i -й ресурс не витрачається повністю в разі реалізації оптимального плану, то відповідне обмеження стає несуттєвим, й оцінка цього ресурсу дорівнює нулю.

2. Якщо для певного оптимального плану двоїстої задачі j -те обмеження виконується як строга нерівність, то оптимальне значення змінної x_j має дорівнювати нулю, тобто якщо $a_{1j}y_1^0 + a_{2j}y_2^0 + \dots + a_{mj}y_m^0 > c_j$, то $x_j = 0$. Виходячи з економічного змісту двоїстих оцінок, останню нерівність можна інтерпретувати як питомі витрати на j -й технологічний процес. Якщо витрати на виробництво певного продукту перевищують прибуток від його реалізації, то його виробництво нерентабельне, тому його не має бути в оптимальному плані ($x_j = 0$).

Аналіз параметричної стійкості розв'язків ЗЛП

Дослідження чутливості та стійкості оптимального плану задачі виробничого планування важливе не лише стосовно зміни обсягів ресурсів. В умовах практичної діяльності підприємства може змінюватись і вартість виробленої продукції, тому не менш важливе питання — стійкість оптимального плану відносно коефіцієнтів цільової функції, тобто параметрична стійкість розв'язку ЗЛП. З економічного погляду ця задача зводиться до визначення меж коливання цін на продукцію підприємства, за яких вироблений план залишається оптимальним.

Зміст цієї проблеми можна проілюструвати геометрично на прикладі задачі з двома змінними. Повернімося до розглянутого раніше прикладу геометричного розв'язання ЗЛП. Оптимальний розв'язок цієї задачі знайдено в точці B , яку утворено перетином двох ліній $x_2 = 300$ й $x_1 + x_2 = 500$. Обмеження, що виконуються в точці оптимуму як рівняння, називають *активними*, усі інші — *пасивними*. Очевидно, що значення оптимуму не зміниться (залишиться в точці B), якщо цільова функція буде обертатися навколо точки B , не перетинаючи ліній активних обмежень, тобто в межах від лінії $x_2 = 300$ до $x_1 + x_2 = 500$ і навпаки. При цьому вектор-градієнт цільової функції буде

змінюватися в межах, визначених нормлями (перпендикулярами) до цих ліній, проведеними через точку B .

З умови оптимальності плану ЗЛП

$$\Delta_j = (\mathbf{c}, A_j) - c_j = z_j - c_j \geq 0$$

можна отримати кількісну оцінку для межі коливання коефіцієнтів цільової функції, за яких оптимальний план не змінюється.

Нехай елемент c_r вектора коефіцієнтів \mathbf{c} змінюється на величину ϵ_r . Можливі два варіанти:

1) якщо вектор A_r не входить до оптимального базису, то необхідна й достатня умова оптимальності плану —

$$\Delta_r = (\mathbf{c}, A_r) - c_r = z_r - c_r \geq 0, \quad (1.17)$$

звідки маємо оцінку для допустимої варіації ϵ_r :

$$\epsilon_r \leq (\mathbf{c}, A_r) - c_r = z_r - c_r;$$

2) якщо вектор A_r входить до базису, то для збереження знайденого оптимального плану потрібно, щоб умова (1.17) виконувалася для всіх небазисних векторів; тоді для варіації ϵ_r має виконуватись умова

$$\epsilon_r a_{rj} \geq c_j - (\mathbf{c}, A_j),$$

де a_{rj} — r -та координата вектора A_j .

Розглянутий приклад дослідження стійкості розв'язку ЗЛП не вичерпує всіх можливих варіантів аналізу зміни параметрів задачі. Ці проблеми вивчають в окремому розділі дослідження операцій, що має назву *параметричного програмування*.

Двоїстий симплекс-метод

Якщо якась ЗЛП має багато обмежень і серед компонент вектора правих частин \mathbf{b} є від'ємні, то для зведення такої задачі до канонічного вигляду може знадобитися багато додаткових і штучних змінних, що у свою чергу призведе до збільшення обсягу обчислень за

симплекс-методом. Тому свого часу було запропоновано так званий двоїстий симплекс-метод, який дає змогу уникнути подібних проблем. Цим методом можна розв'язувати ЗЛП, системи обмежень яких у разі додатного базису мають вільні члени будь-якого знака. За допомогою цього методу можна зменшити кількість перетворень системи обмежень, а також розміри симплексної таблиці.

Крім того, двоїстий симплекс-метод можна застосовувати тоді, коли до системи обмежень додаються нові обмеження після визначення оптимального розв'язку ЗЛП. Саме цю властивість методу застосовують під час розв'язання задач цілочислового програмування. Детальніше це буде розглянуто у відповідному розділі, а зараз розглянемо власне алгоритм методу.

Як відзначалося раніше, між розв'язками пари взаємно двоїстих задач існує тісний зв'язок, і, розв'язуючи одну з них, можна водночас визначити розв'язок іншої. Аналіз симплекс-таблиці з одиничним допоміжним базисом показує, що в стовпцях записано пряму задачу, а в рядках — двоїсту. У звичайному симплекс-методі критерій оптимальності — це невід'ємні значення симплекс-різниць $\Delta_j \geq 0$. Наявність від'ємної симплекс-різниці свідчить про неоптимальність поточного розв'язку. Якщо в розкладі за поточним базисом вектор-стовпець з від'ємною симплекс-оцінкою має від'ємну координату, то план можна поліпшити, змінивши базис. Якщо в цьому векторі немає додатних координат, то це ознака необмеженості цільової функції.

У двоїстому симплекс-методі ситуація аналогічна: критерій оптимальності розв'язку — невід'ємність вільних членів системи обмежень. Якщо серед координат поточного вектора A_0 , який називається *псевдопланом прямої задачі*, є від'ємні елементи, а у відповідному рядку немає жодного від'ємного елемента, то це ознака того, що задача не має розв'язку, бо цільова функція не обмежена на допустимій множині.

Згідно із зазначеними властивостями обчислення за двоїстим симплекс-методом виконують так. Вибирають обмеження з від'ємною правою частиною. Якщо у відповідному рядку симплексної таблиці немає від'ємних елементів, то обчислення закінчують, задача не має оптимального розв'язку. Якщо в цьому рядку є від'ємні елементи $x_{rj} < 0$, то для відповідних векторів A_j обчислюють оцінки

$$\theta_j = \Delta_j / x_{rj},$$

які записують в останньому рядку симплекс-таблиці (під рядком симплекс-різниць $\Delta_j = z_j - c_j$). Вектор A_k , якому відповідає значення $\theta_k = \min \theta_j$, уводять до базису.

Процедуру зміни базису, як і у звичайному симплекс-методі, виконують за правилом виключення Жордана — Гаусса, із тією лише різницею, що провідним (розв'язувальним) є від'ємний елемент x_{rk} .

Зауваження. Якщо вектор правих частин має декілька від'ємних компонент, то як провідний рядок вибирають той, якому відповідає найменше значення b_r .

Після зміни базису перевіряють оптимальність отриманого псевдоплану прямої задачі. Якщо він складається лише з додатних координат, то це її оптимальний план. Тоді за скінченну кількість кроків буде визначено оптимальний план, якщо початкова задача не вироджена, або ж буде виявлено, що задача не має розв'язку.

Контрольні запитання та завдання

1. Які задачі називають задачами лінійного програмування?
2. З яких економічних задач виникають задачі лінійного програмування?
3. Що називають допустимим розв'язком ЗЛП?
4. Який допустимий розв'язок називають опорним?
5. Який розв'язок ЗЛП називають оптимальним?
6. Дайте геометричну інтерпретацію ЗЛП.
7. Яка точка допустимої множини розв'язків називається кутовою?
8. Який зв'язок існує між кутовими точками та базисними розв'язками ЗЛП?
9. Який розв'язок ЗЛП називають виродженим?
10. У яких формах можна записати ЗЛП?
11. Яку форму ЗЛП називають загальною, стандартною? У чому відмінність між ними?
12. Якими засобами можна змінити форму запису ЗЛП?
13. Сформулюйте головну ідею симплексного методу розв'язання ЗЛП.
14. Зазначте послідовність обчислень за симплекс-методом.
15. Як визначають оптимальний розв'язок ЗЛП в симплексному методі?
16. Яка ознака того, що ЗЛП не має оптимального розв'язку?
17. Як визначити початковий опорний розв'язок ЗЛП?
18. Яка роль штучних змінних у розв'язанні ЗЛП?
19. Чим модифікований симплекс-метод відрізняється від звичайного?
20. Як проводиться післяоптимізаційний аналіз розв'язку ЗЛП?
21. Дайте економічну інтерпретацію пари двоїстих ЗЛП.
22. Яка існує відповідність між формами запису двоїстих задач?

23. Який зв'язок існує між розв'язками пари двоїстих ЗЛП?
24. Сформулюйте основні теореми двоїстості.
25. Як застосовують теореми двоїстості для пошуку розв'язку однієї з двоїстих задач за відомим розв'язком іншої?
26. Як за табличним варіантом симплекс-методу визначити розв'язок двоїстої задачі?
27. Чим двоїстий симплекс метод відрізняється від звичайного?
28. Коли застосовують двоїстий симплекс-метод?

2. ТРАНСПОРТНА ЗАДАЧА

Серед ЗЛП особливе місце посідає транспортна задача (ТЗ). Це пов'язано, насамперед, з її практичною значущістю, а також із тим, що специфічні властивості цієї задачі дають змогу значно спростити методи її розв'язання порівняно з тими, що застосовують до звичайних ЗЛП.

Транспортну задачу можна сформулювати так. Нехай є m виробників (постачальників) A_1, A_2, \dots, A_m якоїсь однорідної продукції з максимальними обсягами виробництва a_1, a_2, \dots, a_m і n споживачів B_1, B_2, \dots, B_n цієї продукції з максимальними потребами b_1, b_2, \dots, b_n . Вартість перевезень одиниці продукції c_{ij} від будь-якого виробника A_i до будь-якого споживача B_j відома. Матрицю C з розмірністю $m \times n$, складену з елементів c_{ij} , називають *матрицею вартостей*. Потрібно визначити такі обсяги перевезень x_{ij} від A_i до B_j , щоб мінімізувати транспортні витрати та виконати замовлення всіх споживачів, якщо це принципово можливо.

Математична модель задачі складається з цільової функції

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

множини допустимих значень, заданої нерівностями

$$\begin{aligned} \sum_{j=1}^n x_{ij} &\leq a_i, \quad i = 1, 2, \dots, m, \\ \sum_{i=1}^m x_{ij} &\leq b_j, \quad j = 1, 2, \dots, n, \end{aligned} \tag{2.1}$$

і прямих обмежень

$$x_{ij} \geq 0, \quad i = 1, 2, \dots, m, \quad j = 1, 2, \dots, n.$$

Якщо попит і пропозицію збалансовано, тобто виконується умова

$$\sum_{i=1}^m a_i = \sum_{j=1}^n b_j, \quad (2.2)$$

маємо так звану *закриту* ТЗ. Якщо умова (2.2) не виконується, то задача називається *відкритою*.

Зауваження. Будь-яку відкрити ТЗ можна звести до закритої введенням фіктивного виробника A_{m+1} , коли попит перевищує пропозицію, чи фіктивного споживача B_{n+1} , коли пропозиція перевищує попит. При цьому обсяг продукції фіктивних учасників перевезень становить

$$a_{m+1} = \sum_{j=1}^n b_j - \sum_{i=1}^m a_i$$

чи

$$b_{n+1} = \sum_{i=1}^m a_i - \sum_{j=1}^n b_j,$$

а вартість перевезень від A_{m+1} чи до B_{n+1} уважають нульовою. При цьому в матрицю вартостей C додають рядок з елементами $c_{m+1,j} = 0$, $j = 1, 2, \dots, n$, якщо введено фіктивного виробника, чи стовпець з елементами $c_{i,n+1} = 0$, $i = 1, 2, \dots, m$, якщо введено фіктивного споживача.

Закрита ТЗ має такі властивості:

- 1) вона завжди допустима та має розв'язок;
- 2) серед рівнянь-обмежень (2.1) лише $m + n - 1$ лінійно незалежні;
- 3) якщо в умовах задачі всі числа a_i , $i = 1, 2, \dots, m$, та b_j , $j = 1, 2, \dots, n$, цілі, то оптимальний розв'язок задачі також цілочисловий.

Унаслідок зазначених властивостей пошук розв'язку будь-якої ТЗ починається з перевірки умови (2.2). Якщо вона не виконується, то задачу насамперед зводять до закритого типу, а вже потім застосовують відповідні методи її розв'язання.

Базисні розв'язки ТЗ як різновиду ЗЛП лежать у вершинах допустимої області. Це означає, що будь-який базисний розв'язок складається з додатних і нульових координат, причому додатних коор-

динат у невиродженому розв'язку рівно стільки, скільки лінійно незалежних обмежень у задачі. Якщо розв'язок вироджений, то додатних координат менше, ніж таких обмежень. Отже, із загальних властивостей ЗЛП та другої властивості ТЗ можна дійти висновку, що невироджений базисний розв'язок ТЗ містить рівно $m + n - 1$ ненульових перевезень x_{ij} . У виродженому розв'язку ненульових перевезень менше.

Обчислення, пов'язані з розв'язанням ТЗ, виконують у транспортних таблицях. Вони відрізняються від звичайних симплекс-таблиць тим, що в них заповнюють лише ті клітини (i, j) , у яких перевезення ненульові: $x_{ij} > 0$. Такі клітини називають зайнятими (заповненими), а інші — вільними. Зрозуміло, що невиродженому розв'язку відповідає $m + n - 1$ зайнятих клітин.

Початкова транспортна таблиця містить інформацію про виробників продукції (у першому стовпці), їхні можливості (в останньому), про споживачів продукції (у першому рядку), їхні потреби (в останньому), а також про вартість перевезень c_{ij} , $i = 1, 2, \dots, m, j = 1, 2, \dots, n$, записану в правому верхньому куті внутрішніх клітин таблиці. У центрі $m + n - 1$ клітин записують значення перевезень $c_{ij} > 0$.

Побудувати початковий базисний розв'язок ТЗ значно простіше, ніж для звичайної ЗЛП. Опишемо два найпоширеніші та найчастіше застосовувані методи визначення початкового базисного розв'язку ТЗ.

Метод північно-західного кута. У цьому методі не використовують інформацію про вартість перевезень, а, вибираючи клітину для заповнення, орієнтуються лише на її “географічне” розміщення. Заповнення за цим методом починають із верхньої лівої (північно-західної) клітини. Продукцію першого виробника розподіляють так, щоб максимально задовольнити потреби першого споживача, потім другого і т. д. до повного розподілу всієї виробленої продукції. Якщо попит якогось споживача B_j не задоволено, то продукцію постачають від другого виробника, потім третього і т. д. до повного задоволення попиту. Зрозуміло, що якщо задоволено потреби споживачів B_1, B_2, \dots чи повністю розподілено продукцію виробників A_1, A_2, \dots , то ці виробники та споживачі більше не беруть участі в розподілі. Унаслідок застосування цього методу зайняті клітини транспортної таблиці утворюють характерну східчасту форму: початок “сходів” міститься у верхньому лівому куті таблиці, а кінець — у нижньому правому. Сума перевезень в i -му рядку дорівнює a_i , а в j -му стовпці — b_j .

Отриманий план перевезень, очевидно, є допустимим розв'язком ТЗ. Крім того, він містить не більше ніж $m + n - 1$ заповнених клітин, тобто це базисний розв'язок ТЗ.

Приклад 2.1. Потрібно розподілити продукцію виробників A_1, A_2, A_3 з обсягами виробництва $\mathbf{a} = (10, 15, 7)$ між споживачами B_1, B_2, B_3, B_4 , що мають потреби обсягом $\mathbf{b} = (3, 5, 10, 14)$.

Перевіримо умову (2.2). Маємо $10 + 15 + 7 = 3 + 5 + 10 + 14$. Умова виконується. Виконаємо обчислення в транспортній таблиці, що містить три рядки (за кількістю виробників) і чотири стовпці (за кількістю споживачів), а також інформацію про виробників та їхні можливості (два додаткові стовпці: перший і останній) й інформацію про споживачів (два додаткові рядки: перший і останній). Отже, таблиця має розмірність 5×6 .

Після розподілу продукції за методом північно-західного кута отримуємо такий розв'язок ТЗ:

	B_1	B_2	B_3	B_4	\mathbf{a}
A_1	3	5	2		$a_1 = 10$
A_2			8	7	$a_2 = 15$
A_3				7	$a_3 = 7$
\mathbf{b}	$b_1 = 3$	$b_2 = 5$	$b_3 = 10$	$b_4 = 14$	

Формально обчислення виконують так: для північно-західної клітини (1, 1) вибирають $\min(a_1, b_1)$ і записують його в ній. Оскільки в нашому прикладі $\min(a_1, b_1) = b_1$, то це означає, що попит першого споживача задоволено, і він вибуває з подальшого розподілу, а можливості першого виробника зменшилися на $b_1 = 3$ та становлять $\tilde{a}_1 = 10 - 3 = 7$.

Виведення з розгляду першого споживача означає, що таблиця скорочується на один (перший) стовпець, отже північно-західною стає клітина (1, 2). Для неї також визначають $\min(\tilde{a}_1, b_2) = \min(7, 5) = 5$ і виводять з розгляду другого споживача. При цьому можливості першого виробника скорочуються на b_2 та становлять $\tilde{\tilde{a}}_1 = 7 - 5 = 2$. Для наступної північно-західної клітини (1-3) $\min(\tilde{\tilde{a}}_1, b_3) = \min(2, 10) = 2$. Це означає, що продукцію першого виробника розподілено повністю, але попит третього споживача задоволено лише частково, і він становить $\tilde{\tilde{\tilde{b}}}_3 = 10 - 2 = 8$. Перший рядок виводять із подальшого

розгляду, а північно-західною клітиною стає (2, 3). Для неї $\min(a_2, b_3) = \min(15, 8) = 8$: задоволено попит другого споживача, а можливості другого виробника скорочуються на \tilde{b}_3 та становлять $\tilde{a}_2 = 15 - 8 = 7$. Другий споживач вибуває з подальшого розподілу, північно-західною клітиною стає (2, 4). Для неї $\min(\tilde{a}_2, b_4) = \min(7, 14) = 7$: розподілено продукцію другого виробника, але не задоволено попит четвертого споживача, тобто $\tilde{b}_4 = b_4 - \tilde{a}_2$. Після виведення з розгляду другого рядка північно-західною клітиною стає (3, 4), для якої $a_3 = b_4 = 7$.

Зауважимо, що в цьому методі не розглядають матрицю транспортних витрат C . Тому початковий базисний розв'язок ТЗ, побудований за методом північно-західного кута, зазвичай далекий від оптимального. Наступний метод спирається на інформацію про транспортні витрати й дає початковий розв'язок, який найчастіше ближчий до оптимального, ніж попередній.

Метод мінімального елемента (мінімальної вартості). Ідея цього методу полягає в тому, щоб максимально завантажити комунікації з мінімальними транспортними витратами. Як і в попередньому методі, таблицю заповнюють мінімальними значеннями, що містяться у відповідних клітинах останнього рядка та стовпця, але починають заповнення насамперед із “найдешевших” клітин. Якщо в матриці вартостей декілька однакових елементів, то вибирають будь-який із них.

Метод реалізовано так. На першому кроці визначають мінімальний елемент матриці C . Нехай це буде $c_{i_1 j_1}$. Беруть $x_{i_1 j_1} = \min(a_{i_1}, b_{j_1})$. Можливі два випадки: 1) $a_{i_1} \leq b_{j_1}$ (можливості a_{i_1} виробника A_{i_1} менші, ніж потреби b_{j_1} споживача B_{j_1} ; 2) $a_{i_1} > b_{j_1}$ (навпаки). У першому випадку нулями заповнюють увесь рядок i_1 , крім j_1 -го елемента, а в другому — увесь стовпець j_1 , крім i_1 -го елемента, і з подальших обчислень виключають виробника A_{i_1} (у першому випадку) чи споживача B_{j_1} (у другому). Із матриці C викреслюють відповідний рядок або стовпець, отримують матрицю $C^{(1)}$. Коригують значення відповідних елементів векторів a та b . Беруть

$$\begin{aligned} a_i^{(1)} &= \begin{cases} a_i, & \text{якщо } i \neq i_1, \\ a_i - x_{i_1 j_1}, & \text{якщо } i = i_1; \end{cases} \\ b_j^{(1)} &= \begin{cases} b_j, & \text{якщо } j \neq j_1, \\ b_j - x_{i_1 j_1}, & \text{якщо } j = j_1. \end{cases} \end{aligned} \quad (2.3)$$

На другому кроці виконують описані раніше дії, але стосовно матриці $C^{(1)}$ і векторів $a^{(1)}$ і $b^{(1)}$. Наступні кроки продовжують до повного заповнення транспортної таблиці (чи викреслювання рядків або стовпців матриці C).

Приклад 2.2. Нехай, як і в попередньому прикладі, $\mathbf{a} = (10, 15, 7)$, $\mathbf{b} = (3, 5, 10, 14)$, а матриця транспортних витрат має вигляд

$$C = \begin{pmatrix} 2 & 4 & 3 & 7 \\ 3 & 6 & 3 & 5 \\ 2 & 2 & 5 & 4 \end{pmatrix}.$$

Початковий базисний розв'язок за методом мінімального елемента будують так. Вибирають найменший елемент матриці C . Оскільки їх три, то беруть будь-який із них. Нехай це $(3, 1)$, тобто $i_1 = 3, j_1 = 1$. Тоді $x_{31} = \min(a_3, b_1) = \min(7, 3) = 3$. З матриці C викреслюють перший стовпець і отримують матрицю $C^{(1)}$, коригують значення a_3 та b_1 , тобто беруть $a_3^{(1)} = a_3 - x_{31} = 7 - 3 = 4$, $b_1^{(1)} = b_1 - x_{31} = 0$, а всі інші значення $a^{(1)}$ і $b^{(1)}$ залишають такими, як і в попередніх векторах \mathbf{a} та \mathbf{b} .

Мінімальний елемент матриці $C^{(1)}$ (за початковим розміщенням) — $(3, 2)$. Тоді $x_{32} = \min(a_3^{(1)}, b_2^{(1)}) = \min(4, 5) = 4$. З матриці $C^{(1)}$ викреслюють останній рядок, отримують матрицю $C^{(2)}$, обчислюють значення векторів $a^{(2)}$ та $b^{(2)}$ за формулами, аналогічними (2.3).

Пропонований метод — модифікація методу північно-західного кута. Він також дає змогу визначити допустимий базисний розв'язок. Хоча застосування матриці транспортних витрат робить цей розв'язок досить близьким до оптимального, але оптимальний визначають за методом потенціалів.

Метод потенціалів розв'язання транспортної задачі. Як і для звичайної ЗЛП, двоїсті змінні посідають значне місце в дослідженні та розв'язанні ТЗ. Такі змінні в ТЗ називають *потенціалами* та розбивають їх на дві групи: $u_i, i = 1, 2, \dots, m$, — потенціали пунктів відправлення A_1, A_2, \dots, A_m , $v_j, j = 1, 2, \dots, n$, — потенціали пунктів призначення B_1, B_2, \dots, B_n .

Двоїстий критерій оптимальності ТЗ, на якому ґрунтується метод потенціалів, формують так: допустимий базисний розв'язок ТЗ оптимальний тоді й лише тоді, коли існують такі потенціали $u_i, i = 1, 2, \dots, m, v_j, j = 1, 2, \dots, n$, що їх сума для всіх додатних координат

цього розв'язку дорівнює вартості перевезень і менша за відповідну вартість нульових координат:

$$\begin{cases} u_i + v_j = c_{ij}, & \text{якщо } x_{ij} > 0, \\ u_i + v_j \leq c_{ij}, & \text{якщо } x_{ij} = 0. \end{cases}$$

Цей критерій можна сформулювати інакше за допомогою поняття відносних оцінок. Величину $\Delta_{ij} = c_{ij} - (u_i + v_j)$, $i = 1, 2, \dots, m$, $j = 1, 2, \dots, n$, назовемо відносною оцінкою змінної x_{ij} . Тоді, згідно з двоїтим критерієм оптимальності, базисний розв'язок $X\{x_{ij}\}$, $i = 1, 2, \dots, m$, $j = 1, 2, \dots, n$, оптимальний тоді й лише тоді, коли відносні оцінки $\Delta_{ij} = 0$ для базисних (зайнятих) клітин транспортної таблиці та $\Delta_{ij} \geq 0$ для небазисних (вільних) клітин таблиці.

Викладемо сутність методу потенціалів.

1. Як перше наближення до оптимального розв'язку вибирають будь-який початковий базисний розв'язок (побудований методом північно-західного кута, мінімального елемента чи іншим методом).

2. Визначають потенціали так, щоб у кожній базисній (ненульовій) клітині виконувалась умова $\Delta_{ij} = 0$, тобто складають систему рівнянь

$$u_i + v_j = c_{ij} \tag{2.4}$$

для всіх базисних клітин $x_{ij} > 0$. Зауважимо, що така система містить $m + n - 1$ рівнянь (за кількістю базисних клітин) й $m + n$ невідомих, тому (як відомо з матричної алгебри) має безліч розв'язків. Щоб визначити хоча б один із них, одну зі змінних довизначають (наприклад, беруть $u_1 = 0$), а всі інші знаходять із зазначеної системи.

3. За знайденими потенціалами обчислюють відносні оцінки небазисних (нульових) клітин. Якщо для усіх цих клітин $\Delta_{ij} \geq 0$, то даний базисний розв'язок оптимальний. Якщо ж існують небазисні клітини, для яких $\Delta_{ij} < 0$, то цей розв'язок можна поліпшити, тобто перейти до іншого базисного розв'язку з меншою вартістю перевезень. Як і в звичайному симплекс-методі, зміна базису полягає в тому, щоб до базису ввести змінну, яка має "найгіршу" відносну оцінку, і вивести одну з поточних базисних змінних.

Очевидно, що до базису вводять змінну з найменшою від'ємною відносною оцінкою $\Delta_{ij} < 0$. Нехай це клітина (i_0, j_0) . Визначимо, яку з базисних клітин слід вивести з базису. Для цього з клітин транспортної таблиці утворимо цикл (замкнений ланцюг) так, щоб він містив клітину (i_0, j_0) (небазисну, нульову) і деякі базисні клітини. Позначимо клітини циклу так: небазисну — знаком “+”, а інші (базисні) — знаком “-” чи “+”, але так, щоб у кожному рядку та стовпці транспортної таблиці була однакова кількість різних знаків, що чергуються між собою. Кожну наступну клітину для позначення відповідним знаком вибирають лише в тому самому рядку чи стовпці, де вже є позначена. Починають і закінчують цикл у небазисній клітині (i_0, j_0) .

Серед базисних клітин, позначених знаком “-”, виберемо найменше перевезення $x_{ij} = x_{\min}^-$. Збільшимо обсяг перевезень на $\theta = x_{\min}^-$ у клітинах, позначених знаком “+”, і зменшимо на ту саму величину обсяг перевезень у клітинах, позначених знаком “-”. Після такого перерозподілу вантажу одна з базисних клітин стане вільною, тобто вийде з базису, а клітина (i_0, j_0) ввійде до нього й міститиме перевезення $x_{i_0 j_0} = \theta > 0$.

4. Отриманий новий базисний розв'язок перевіряють на оптимальність. Процедура виконується доти, доки для поточного базисного розв'язку не виконається критерій оптимальності.

Метод потенціалів у невиродженій задачі дає змогу визначити оптимальний розв'язок ТЗ за скінченну кількість кроків. У виродженій задачі, коли кількість заповнених клітин менша ніж $m + n - 1$, можливе зациклювання. Для запобігання йому в одну з вільних клітин вводять дуже мале фіктивне перевезення $\epsilon > 0$, яке не впливає на загальний план перевезень, але формально задовольняє вимогу методу потенціалів про невиродженість базисного розв'язку, тобто про $m + n - 1$ заповнених клітин. Клітину для заповнення фіктивним перевезенням вибирають так, щоб система (2.4) мала розв'язок, за умови, що одну з невідомих визначають довільно.

Приклад 2.3. Для розглянутої раніше задачі (приклади 2.1, 2.2) знайдемо оптимальний розв'язок методом потенціалів. Як початковий базисний візьмемо розв'язок, отриманий методом північно-західного кута.

Для базисних клітин $(1,1)$, $(1,2)$, $(1,3)$, $(2,3)$, $(2,4)$, $(3,4)$ обчислимо потенціали u_i , $i = 1, 2, \dots, m$, v_j , $j = 1, 2, \dots, n$, за співвідношеннями (2.4). Розглянемо систему

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

Вона складається з шести рівнянь і містить сім невідомих $u_1, u_2, u_3, v_1, v_2, v_3, v_4$, одну з яких потрібно до визначити. Нехай $u_1 = 0$; тоді $v_1 = 2; v_2 = 4; v_3 = 3; u_2 = 0; u_4 = 5; u_3 = -1$.

Занесемо ці дані до таблиці

	B_1	B_2	B_3	B_4	a_i	u_i
A_1	3	5	2		10	0
A_2			8	7	15	0
A_3				7	7	-1
b_j	3	5	10	14		
v_j	2	4	3	5		

Для незаповнених (нульових, небазисних) клітин обчислимо відносні оцінки за формулою

$$\Delta_{ij} = c_{ij} - (u_i + v_j).$$

Отримаємо

$$\Delta_{14} = c_{14} - (u_1 + v_4) = 7 - (0 + 5) = 2;$$

$$\Delta_{21} = c_{21} - (u_2 + v_1) = 3 - (0 + 2) = 1;$$

$$\Delta_{22} = c_{22} - (u_2 + v_2) = 6 - (0 + 4) = 2;$$

$$\Delta_{31} = c_{31} - (u_3 + v_1) = 2 - (-1 + 2) = 1;$$

$$\Delta_{32} = c_{32} - (u_3 + v_2) = 2 - (-1 + 4) = -1;$$

$$\Delta_{33} = c_{33} - (u_3 + v_3) = 5 - (-1 + 3) = 3.$$

Зауваження. Відносні оцінки, як і вартість перевезень, можна також занести до транспортної таблиці. Зазвичай значення Δ_{ij} запи-

сують у лівому верхньому куті відповідної клітини, вартість перевезень c_{ij} — у верхньому правому, а обсяг перевезень x_{ij} — у центрі клітини. Тоді найповніша транспортна таблиця має вигляд

	B_1	B_2	B_3	B_4	a_i	u_i
A_1	2 3	4 5 ⁻	3 2 ⁺	2 7	10	0
A_2	1 3	2 6	3 8 ⁻	5 7 ⁺	15	0
A_3	1 2	-1 2 +	3 5	4 7 ⁻	7	-1
b_j	3	5	10	14		
v_j	2	4	3	5		

Єдина вільна клітина в таблиці з від'ємною відносною оцінкою — (3, 2). Позначимо її знаком “+” і поєднаємо з базисними клітинами циклом. У другому стовпці таблиці є лише одна базисна клітина — (1, 2). Позначимо її знаком “-”, бо сусідня клітина (3, 2) в циклі вже має знак “+”. У першому рядку є ще дві базисні клітини: (1, 1) і (1, 3), причому (1, 1) не можна включати до циклу, тому що в першому стовпці немає базисної клітини, яку можна було б позначити протилежним знаком. Тому знак “+” поставимо в клітині (1, 3). Тоді (2, 3) отримає знак “-”, (2, 4) — знак “+”, (3, 4) — знак “-”, і цикл завершиться, як і почався, у клітині (3, 2). Кількість плюсів і мінусів у кожному рядку та стовпці однакова. У клітинах, позначених знаком “-”, визначимо найменше перевезення $\theta = \min(5, 8, 7) = 5$ і, перевантажуючи клітини транспортної таблиці за зазначеним циклом, отримуємо новий базисний розв'язок:

	B_1	B_2	B_3	B_4	a_i	u_i
A_1	3	5 - 5 = 0	2 + 5 = 7		10	0
A_2			8 - 5 = 3	7 + 5 = 12	15	0
A_3		0 + 5 = 5		7 - 5 = 2	7	-1
b_j	3	5	10	14		
v_j	2	4	3	5		

Обчислюючи, як і раніше, потенціали для нового базисного розв'язку, а потім відносні оцінки для відповідних небазисних змінних, переконаємось, що даний розв'язок задовольняє критерій оптимальності, тобто відносні оцінки для всіх клітин транспортної таблиці невід'ємні: $\Delta_{ij} \geq 0$ для всіх значень $i = 1, 2, 3, j = 1, 2, 3, 4$.

Зауваження. Оскільки критерій оптимальності перевіряють на підставі відносних оцінок небазисних клітин таблиці, то можна не обчислювати значення потенціалів на кожній ітерації, а лише шукати значення відносних оцінок для нового плану, виходячи з відносних оцінок попереднього плану. Такі обчислення зазвичай виконують у матричному вигляді. Для цього в матриці оцінок підкреслюють усі нульові елементи, що відповідають базисним клітинам, і визначають її найменший від'ємний елемент. Рядок, якому належить цей елемент, викреслюють. Якщо в цьому рядку є підкреслені (нульові) елементи, то викреслюють відповідні стовпці. Якщо у викреслених стовпцях є підкреслені елементи, то викреслюють відповідні рядки і т. д., поки таку процедуру можна продовжувати. Тоді нову матрицю оцінок визначають за таким правилом: від усіх елементів викреслених рядків віднімають мінімальну відносну оцінку та додають її до всіх елементів викреслених стовпців. На новій ітерації матрицю оцінок Δ^2 обчислюють аналогічно на підставі Δ^1 .

У нашому прикладі матриця Δ^0 , що містить значення відносних оцінок, отриманих для початкового базисного розв'язку, має вигляд

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

Знайдемо нову матрицю оцінок Δ^1 . Мінімальна від'ємна відносна оцінка дорівнює -1 і належить третьому рядку. Викреслимо цей рядок. У ньому є нульова відносна оцінка, що відповідає базисній клітині (3, 4). Викреслимо четвертий стовпець. У ньому є нульовий елемент, що відповідає базисній клітині (2, 4). Викреслимо другий рядок, який, у свою чергу, містить базисну клітину (2, 3) (з нульовою відносною оцінкою). Це означає, що слід викреслити третій стовпець. Він також містить базисну клітину (1, 3), яка показує, що слід викреслити й перший рядок. У ньому є ще дві базисні клітини; відповідні їм стовпці також треба викреслити. Однак викреслення другого стовпця

приведе до того, що задачу виключення від'ємної відносної оцінки не буде розв'язано: нова матриця оцінок збігатиметься з попередньою матрицею. Тому викреслимо лише перший стовпець. Тоді викресленими виявляться всі рядки та всі стовпці, крім другого, а це означає, що зміниться лише цей стовпець. Матриця оцінок набуде вигляду

$$\Delta^1 = \begin{pmatrix} 0+1-1 & 0+1 & 0+1-1 & 2+1-1 \\ 1+1-1 & 2+1 & 0+1-1 & 0+1-1 \\ 1+1-1 & -1+1 & 3+1-1 & 0+1-1 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 2 \\ 1 & 3 & 0 & 0 \\ 1 & 0 & 3 & 0 \end{pmatrix}.$$

Усі її елементи додатні, отже побудований розв'язок ТЗ оптимальний.

Такий самий результат можна отримати, розв'язавши систему (2.4) й обчисливши відносні оцінки небазисних клітин цього розв'язку ТЗ.

Описані перетворення для матриць із малою розмірністю, можливо, не здаються дуже ефективними, однак у разі великої розмірності системи (2.4) вони виявляються ефективнішими, ніж розв'язування цієї системи.

Багато практичних задач, пов'язаних із плануванням перевезень, не вкладаються в межі розглянутої задачі, бо ускладнені додатковими обмеженнями чи взагалі суттєво складніші за постановкою.

Наприклад, реальні магістралі, що зв'язують виробників продукції з її споживачами, можуть мати обмежені (у певний період) пропускні здатності. Ці обмеження можуть завадити виконанню знайденого оптимального плану перевезень, тому бажано додати їх до умов задачі. Математична модель задачі з обмеженими пропускними здатностями магістралей складається, як і раніше, з цільової функції

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

обмежень на продукцію

$$\sum_{j=1}^n x_{ij} \leq a_i, \quad i = 1, 2, \dots, m, \quad \sum_{i=1}^m x_{ij} \leq b_j, \quad j = 1, 2, \dots, n,$$

та прямих обмежень на перевезення

$$0 \leq x_{ij} \leq d_{ij}, \quad i = 1, 2, \dots, m, \quad j = 1, 2, \dots, n.$$

Тут d_{ij} — пропускна здатність магістралі (i, j) , тобто максимальний обсяг продукції, який можна перевезти нею за обмежений проміжок часу.

Така задача може виявитися нерозв'язною, якщо пропускна здатність магістралей, що ведуть до j -го споживача, менша за його потреби, тобто $\sum_{i=1}^m d_{ij} < b_j$.

Для розв'язання задач з обмеженими пропускними здатностями застосовують *спеціальний метод потенціалів*, який за скінченну кількість кроків (ітерацій) дає змогу визначити оптимальний план перевезень або виявити нерозв'язність такої задачі.

Початкова транспортна задача може бути ускладнена також наявністю проміжних транспортних вузлів, у яких обробляють вантажі (наприклад, перевантажують на інший вид транспорту чи на склади), або ж браком магістралей, що з'єднують окремих виробників і споживачів. Такі задачі розв'язують специфічними методами.

Характер моделі задачі принципово змінюється навіть у найпростішому випадку, якщо критерій оптимальності — час, потрібний для задоволення всіх потреб споживачів. Задача, у якій треба мінімізувати час перевезень, називається ТЗ за критерієм часу. Її математична модель нелінійна внаслідок нелінійності цільової функції. Якщо замість матриці вартості перевезень $C = (c_{ij})$ задано матрицю $T = (t_{ij})$, елементи якої — час перевезень, причому кожне значення t_{ij} не залежить від обсягу продукції, яку перевозять, то цільова функція такої задачі набуває вигляду

$$\max t_{ij} \rightarrow \min.$$

Тут $\max t_{ij}$ означає максимальне значення серед елементів t_{ij} , що відповідають ненульовим перевезенням $x_{ij} > 0$ (для $x_{ij} = 0$ вважаємо $t_{ij} = 0$) розглянутого допустимого плану перевезень. Зрозуміло, що за час $\max t_{ij}$ буде виконано всі перевезення. Потрібно добрати такий допустимий план, для якого цей час мінімальний.

Розв'язання такої задачі можна звести до послідовного розв'язання декількох ЗЛП чи застосувати алгоритм, що дає змогу поступово поліпшувати початковий опорний план за правилами, подібними до методу потенціалів.

Транспортна задача значно ускладнюється, якщо потрібно скласти раціональний план перевезень не одного, а кількох видів продукції. Ще складнішою виявляється задача, у якій потрібно розподілити перевезення між різними видами транспортних засобів, причому вартість перевезень одиниці продукції від i -го виробника до j -го споживача залежить не тільки від виду продукції, але й від використаного для цього виду транспорту. Такі задачі називаються *багатоіндексними* ТЗ. Збільшення кількості індексів, що відповідають різним видам продукції та різним видам транспорту, значно збільшує кількість обчислень, потрібних для розв'язання таких задач і потребує застосування методів, у яких враховано зазначені особливості.

На завершення зауважимо, що багато задач транспортного типу можна розв'язувати методами, що ґрунтуються на математичному аналізі властивостей відповідних транспортних мереж.

Графічне зображення транспортної мережі (так званої *граф*) — набір вершин (транспортних вузлів) і дуг (комунікацій), що з'єднують відповідні пункти.

Наприклад, граф $\Gamma = \{I, U\}$ можна описати *множиною вершин* $I = \{1; 2; 3; 4; 5\}$ і *множиною дуг* $U = \{(1; 2), (1; 3), (2; 1), (2; 4), (4; 3), (3; 5), (5; 2)\}$. Оскільки транспортні перевезення мають певний напрямок, то пари вершин упорядковані, і граф називається *орієнтованим* (напрямленим).

Зауваження. Якщо напрямок руху не зазначено, то дуги графа називаються *ребрами*, а граф — *неорієнтованим*.

“Транспортною” мовою послідовність взаємопов'язаних вершин називають *маршрутом*, а напрямлений маршрут — *шляхом*.

З поняттям графу тісно пов'язане поняття *мережі*. Так називають граф, елементам якого поставлено у відповідність певні параметри: вершинам — d_i , дугам — r_{ij} . На практиці величини d_i інтерпретують як обсяги виробництва ($d_i > 0$) чи обсяги споживання ($d_i < 0$) певного однорідного продукту; r_{ij} — обмеження на пропускні здатності комунікацій, що пов'язують пункти i, j . Параметром дуги може бути також вартість перевезень c_{ij} одиниці продукції з пункту i в пункт j .

Поняття *поток*, яке вводять на графах, узагальнює поняття допустимих транспортних перевезень. Як і фізичний аналог потоку

рідини у з'єднаних сосудах, потік на графі має властивості збереження та неперервності. Залежно від критеріїв оптимальності на мережі ставлять задачі про оптимальний потік (якщо оптимізують вартість перевезень), найкоротший шлях (коли з багатьох варіантів маршруту вибирають оптимальний), максимальний потік (коли потрібно визначити максимальну пропускну спроможність мережі) та ін.

Методи розв'язання транспортних (та інших) задач, що ґрунтуються на теорії графів, часто виявляються ефективнішими від розглянутих матричних.

Детальніше з методами розв'язання транспортних задач яку в матричній, так і мережній постановці можна ознайомитися в [3].

Контрольні запитання та завдання

1. Які задачі називають транспортними?
2. Які ТЗ називають закритими, а які відкритими?
3. Назвіть основні властивості закритої ТЗ.
4. Як перейти від відкритої ТЗ до закритої?
5. Який розв'язок ТЗ називають базисним?
6. Який базисний розв'язок ТЗ називають невиродженим, а який виродженим?
7. У якому вигляді шукають розв'язок ТЗ?
8. Які методи застосовують для побудови початкового опорного розв'язку ТЗ?
9. Як будують початковий розв'язок ТЗ методом північно-західного кута?
10. Як будують початковий розв'язок ТЗ методом мінімальної вартості?
11. Який метод застосовують для побудови оптимального розв'язку ТЗ?
12. Як визначають потенціали пунктів постачання та споживання?
13. Яка ознака оптимального розв'язку ТЗ?
14. Які економічні задачі можна розв'язувати як транспортні?
15. Назвіть інші моделі ТЗ.

3. ЦІЛОЧИСЛОВІ ТА ДИСКРЕТНІ ЗАДАЧІ ЛІНІЙНОГО ПРОГРАМУВАННЯ

Серед практично важливих задач пошуку умовного екстремуму лінійної функції важливе місце посідають задачі з вимогою цілочисловості всіх змінних або їх частини.

Якщо в задачах вибору оптимального асортименту мова йде про масове виробництво (гайок, болтів, цвяхів та ін.), умова цілочисловості кількості виробів не принципова, бо округлення великих чисел до цілих не надто змінює загальний результат. Однак для дрібно-серійного чи унікального обладнання (літаків, ракет і т. ін.) ця умова дуже суттєва.

Перш ніж формально означити задачі цілочислового програмування, розглянемо декілька прикладів.

Задача про призначення. Нехай є n видів робіт і n кандидатів для їх виконання. Припускають, що кожного кандидата можна призначити лише на один вид робіт. Позначимо як c_{ij} ефективність призначення i -го кандидата на j -й вид робіт. Потрібно розподілити кандидатів на роботу так, щоб загальний дохід від виконаних робіт був максимальним.

Нехай змінні задачі набувають лише двох значень: $x_{ij} = 1$, якщо i -го кандидата призначено для виконання j -ї роботи, а не то $x_{ij} = 0$. Тоді математична модель задачі про призначення має такий вигляд: максимізувати

$$z = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij}$$

для обмежень

$$\sum_{j=1}^n x_{ij} = 1, \quad i = 1, 2, \dots, n; \quad \sum_{i=1}^n x_{ij} = 1, \quad j = 1, 2, \dots, n,$$

де $x_{ij} = 1$ чи $x_{ij} = 0$.

Задача про комівояжера. Торговий агент має відвідати лише по одному разу n пунктів, вартість переїзду між пунктами відома. Потрібно скласти такий маршрут, щоб загальні витрати під час переїздів були якомога меншими.

Математична модель цієї задачі майже повністю збігається з моделлю попередньої. Її розв'язок — послідовність пунктів, через які проходить маршрут комівояжера. Для відображення цього маршруту застосовують змінні, які можуть набувати лише двох значень: $x_{ij} = 1$, якщо комівояжер переїжджає з i -го пункту в j -й; $x_{ij} = 0$, якщо переїзду між пунктами немає.

Тоді математична модель задачі про комівояжера має такий вигляд: мінімізувати

$$z = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij}$$

для обмежень

$$\sum_{j=1}^n x_{ij} = 1, \quad i = 1, 2, \dots, n; \quad \sum_{i=1}^n x_{ij} = 1, \quad j = 1, 2, \dots, n,$$

де $x_{ij} = 1$ чи $x_{ij} = 0$.

Задача про рюкзак. Мандрівник, який збирається в подорож, має зібрати свій рюкзак (валізу чи саквояж), що має m обмежень b_i ($i = 1, 2, \dots, m$) за розміром, вагою, об'ємом та ін. У подорожі може знадобитися деяка кількість предметів n найменувань, кожен з яких має певні характеристики за тими самими ознаками, що й сам рюкзак. Нехай a_{ij} — i -та ($i = 1, 2, \dots, m$) характеристика предмета j -го найменування. Кожен із потрібних у подорожі предметів має цінність c_j , $j = 1, 2, \dots, n$. Потрібно скласти рюкзак так, щоб його цінність була максимальною. Зрозуміло, що неможливо брати речі частинами, тому змінні x_j — кількість предметів j -го найменування —

мають набувати лише цілих значень. Математична модель цієї задачі має такий вигляд: максимізувати

$$z = \sum_{j=1}^n c_j x_j$$

для обмежень

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

$$x_{ij} \geq 0, \quad x_{ij} \text{ — цілі, } j = 1, 2, \dots, n.$$

Зауважимо, що в початковій постановці такої задачі мова йшла про завантаження бомбардувальників різних типів бомбовим запасом для максимізації загального ефекту системи бойових операцій.

Задача про вибір транспортних засобів. Нехай яесь транспортне підприємство обслуговує n маршрутів із приблизно однаковою кількістю пасажирів на кожному. Кожен транспортний засіб має свої технічні характеристики (місткість, витрати пального та ін.), які визначають витрати підприємства для обслуговування наявних транспортних засобів. Прибуток підприємства залежить від кількості проданих квитків. Потрібно вибрати для кожного маршруту такі транспортні засоби та в таких кількостях, щоб перевезти всіх пасажирів і максимізувати прибуток підприємства.

Формалізуємо цю задачу. Нехай j -м маршрутом можна перевезти $b_1^{(j)}$ пасажирів. Перевезення можна виконати m різними типами транспортних засобів. Для кожного (i -го) з них відомі такі характеристики:

- 1) $a_i^{(1)}$ — місткість (кількість місць);
- 2) $a_i^{(2)}$ — чисельність обслуги;
- 3) $a_i^{(3)}$ — витрати пального за певний період часу;
- 4) c_{ij} — прибуток від використання i -го засобу на j -му маршруті.

Потрібно вибрати транспортний парк для кожного маршруту, який забезпечить перевезення всіх пасажирів, максимізує прибуток підприємства, але не перевищить можливостей підприємства щодо витрат пального (не більше ніж b_3) і чисельності обслуги (не більше ніж b_2 працівників).

Математична модель цієї задачі має такий вигляд: максимізувати

$$F(x) = \sum_{j=1}^n \sum_{i=1}^m c_{ij} x_{ij}$$

для обмежень

$$\sum_{i=1}^m \sum_{j=1}^n a_i^{(k)} x_{ij} \leq b_k, \quad k = 1, 2, 3,$$

$$x_{ij} \geq 0, x_{ij} \text{ — цілі, } \quad i = 1, 2, \dots, m, \quad j = 1, 2, \dots, n.$$

Транспортна задача з фіксованими доплатами. Вона відрізняється від звичайної ТЗ тим, що витрати на перевезення однорідного вантажу з пункту A_i в пункт B_j визначають так:

$$\tilde{c}_{ij}(x_{ij}) = \begin{cases} 0, & \text{якщо } x_{ij} = 0, \\ c_{ij}x_{ij} + d_{ij}, & \text{якщо } x_{ij} > 0, \end{cases}$$

де c_{ij} — витрати на перевезення одиниці вантажу; d_{ij} — фіксована доплата за оренду транспортних засобів.

Із такими припущеннями функція сумарних витрат має розриви, що значно ускладнює її мінімізацію. Тому вводять додаткові бінарні змінні (що набувають лише двох значень: 0 або 1), які дають змогу усунути розриви функції витрат і звести початкову задачу до *частково цілочислової*. Так називають задачі, у яких умову цілочисловості накладено лише на частину змінних. Якщо ж змінні задачі можуть набувати лише скінченної кількості значень (не обов'язково цілих), то виникають задачі *дискретного програмування*.

Зазначені задачі поділяють на такі класи:

- 1) із неподільностями (про рюкзак, вибір транспортних засобів);
- 2) екстремальні комбінаторні (про призначення, про комівояжера);
- 3) із розривними цільовими функціями (транспортна з фіксованими доплатами).

Узагальнюючи наведені приклади, можна отримати таку *модель задачі цілочислового програмування*: максимізувати

$$F(x) = \sum_{j=1}^n c_j x_j$$

за умов

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

$$x_j \geq 0, \quad x_j \text{ — цілі, } j = 1, 2, \dots, n.$$

Наведена модель відрізняється від звичайної ЗЛП лише умовою цілочисловості, накладеною на змінні x_j . Однак ця умова спричиняє принципову складність розв'язання задач цілочислового програмування. Це пов'язано з тим, що в більшості випадків неможливо замінити дискретну задачу її неперервним аналогом — звичайною ЗЛП, розв'язок якої можна було б округлити до цілих значень.

Приклад 3.1. Розглянемо задачу

$$x_1 - 3x_2 + 3x_3 \rightarrow \max,$$

$$\begin{cases} 2x_1 + x_2 - x_3 \leq 4, \\ 4x_1 - 3x_2 \leq 2, \\ -3x_1 + 2x_2 + x_3 \leq 3, \\ x_j \geq 0, \quad x_j \text{ — цілі, } j = 1, 2, 3. \end{cases}$$

Якщо розв'язувати її без урахування вимоги цілочисловості змінних, тобто як звичайну ЗЛП, то отримаємо розв'язок $x^* = (1/2; 0; 9/2)$. Насправді ж розв'язок цілочислової задачі — $x_{ц}^* = (2; 2; 5)$. Очевидно, що жоден із варіантів округлення розв'язку допоміжної ЗЛП не дасть навіть допустимого розв'язку наведеної задачі цілочислового програмування.

Геометрично області допустимих розв'язків звичайної ЗЛП та цілочислової задачі можна подати так (рис. 3.1).

Якщо у звичайній ЗЛП перейти від допустимої області D до допустимої області цілочислової задачі $D_{ц}$, то розв'язок звичайної ЗЛП буде цілочисловим. На жаль, безпосередній перехід від цілочислової задачі до ЗЛП практично нездійснений, оскільки побудувати область $D_{ц}$ дуже важко. Цю задачу можна розв'язати частково, послі-

яке відповідає такому β_i , що $\{\beta_i\} \neq 0$. Правдиве таке твердження.

Теорема 3.1. Додаткове лінійне обмеження (3.3) — правильний відтин для задачі цілочислового ЛП (3.1).

Доведено, що це обмеження не виконується для будь-якого нецілочислового базисного плану x , але виконується для його цілих значень. Для реалізації правильного відтину до системи обмежень, отриманої на останній ітерації, додають умову

$$- \sum_{j=m+1}^n \{\alpha_{ij}\}x_j + x_{n+1} = -\{\beta_i\},$$

де x_{n+1} — додаткова змінна, завдяки якій нерівність перетворюється в строгу рівність. У цільовій функції їй відповідає нульовий коефіцієнт. Таблиця при цьому розширюється на один рядок і один стовпець. Для розв'язання цієї задачі застосовують двоїстий симплекс-метод.

Оскільки в початковому псевдоплані новоутвореної задачі тільки одна від'ємна компонента, то з базису слід вивести вектор, що відповідає цій компоненті, тобто A_{n+1} . Рядок, що відповідає цій базисній змінній, провідний. Для визначення змінної, яку слід увести до базису, обчислюють значення $\theta_j = -\Delta_j / \alpha_{n+1,j}$, де $\alpha_{n+1,j} = -\{\alpha_{ij}\} \neq 0$, і записують їх в останньому рядку таблиці. Провідному стовпцю відповідає найменше додатне значення θ_j . Провідним елементом таблиці буде той, що стоїть на перетині провідного рядка та провідного стовпця.

Зауваження. Позаяк за означенням дробова частина числа — це відстань до найближчого цілого числа, що лежить на числовій осі лівіше даного, то $\{z\}$ більша нуля як для додатних, так і для від'ємних значень z . Наприклад, $\{-4/3\} = -4/3 - (-2) = 2/3$. Тому провідний елемент також завжди від'ємний.

Процедуру зміни базису зазвичай виконують методом виключення Жордана — Гауса. Якщо отриманий оптимальний розв'язок виявиться цілочисловим, то обчислення закінчують, а не то знову проводять правильне відтинання.

Двоїстий симплекс-метод — це основа методу Гоморі, бо він дає змогу враховувати додаткові обмеження (правильні відтини) і переходити від поточного псевдоплану до нового оптимального плану.

Доведено, що наведений алгоритм скінченний. Це означає, що на якомусь кроці (ітерації) буде знайдено цілочисловий план або виявлено, що допустимих цілочислових планів для даної задачі не існує.

Зауваження. Виконуючи розрахунки на комп'ютері, слід зважати на похибки округлення, тому що внаслідок особистостей машинної арифметики жоден план не може бути цілочисловим. Крім того, нагромадження похибок може порушити порядок обчислень і взагалі “відвести” від оптимального розв'язку початкової задачі.

Приклад 3.2. Розв'язати цілочислову ЗЛП

$$x_1 + x_2 - 2x_3 \rightarrow \min$$

$$\begin{cases} x_1 + 5x_3 = 4, \\ x_2 + 5x_3 = 1, \\ x_j \geq 0, \quad x_j \text{ — цілі, } j = 1, 2, 3. \end{cases}$$

Розв'яжемо допоміжну ЗЛП, отриману з початкової задачі відкиданням умови цілочисловості змінних $x_j, j = 1, 2, 3$. Виконаємо обчислення в симплекс-таблиці.

Крок	$c_{\text{баз}}$	Базис	A_0	1	1	2	θ
				A_1	A_2	A_3	
0	-1	x_1	4	1	0	5	4/5
	1	x_2	1	0	1	5	1/5
		Δ	-3	0	0	2	
1	-1	x_1	3	1	-1	0	
	-2	x_3	1/5	0	1/5	1	
		Δ	-17/5	0	-2/5	0	

На останньому кроці обмеження допоміжної задачі мають вигляд

$$\begin{cases} x_1 - x_2 = 3, \\ 1/5 x_2 + x_3 = 1/5. \end{cases}$$

За другим обмеженням, що відповідає нецілому значенню правої частини, будемо правильний відтин: до розв'язаної ЗЛП додаємо обмеження

$$1/5 - 1/5x_2 \leq 0, \quad \text{або} \quad -1/5x_2 \leq -1/5.$$

Отже, нова допоміжна ЗЛП містить додаткове обмеження

$$-1/5x_2 + x_4 = -1/5,$$

де x_4 — додаткова змінна, яка зводить останню нерівність до канонічного вигляду, тобто перетворює її на рівність. Коефіцієнт при x_4 в цільовій функції — $c_4 = 0$.

Виконаємо обчислення двоїтим симплекс-методом.

Крок	$c_{\text{баз}}$	Базис	A_0	-1	1	-2	0
				A_1	A_2	A_3	A_4
0	-1	x_1	3	1	-1	0	0
	-2	x_3	1/5	0	1/5	1	0
	0	x_4	-1/5	0	-1/5	0	1
		Δ θ	-17/5	0 0	-2/5 2	0 0	0 0
1	-1	x_1	4	1	0	0	-1
	2	x_3	0	0	0	1	1/5
	1	x_2	1	0	1	0	-1

Оскільки у векторі A_0 компонента $\alpha_{04} = \beta_4 = -1/5$ від'ємна, то змінну x_4 потрібно вивести з базису. Тому провідний рядок — той, що відповідає базисній змінній x_4 , провідний стовпець — A_2 , бо тільки йому відповідає додатне значення $\theta_2 = \Delta_2 / \alpha_{42} = 2$, а провідний елемент — $\alpha_{42} = -1/5$ (позаяк ми розв'язуємо задачу мінімізації, а не максимізації, то для обчислення θ_2 взято Δ_2 , а не $-\Delta_2$).

Отримано оптимальний цілочисловий розв'язок $x^* = (4; 1; 0; 0)$ допоміжної ЗЛП. Це означає, що $x^* = (4; 1; 0; 0)$ — оптимальний розв'язок початкової цілочислової задачі.

У загальному випадку обчислення за двоїтим симплекс-методом закінчують після того, як у стовпці вільних членів A_0 не залишиться від'ємних елементів. Якщо A_0 містить лише цілі числа, то отримано розв'язок цілочислової задачі, а не то знову виконують правильне відтинання.

Ще раз зауважимо, що в практичних обчисленнях збіжність цього методу дуже повільна, а з підвищенням розмірності реалізація його алгоритму значно ускладнюється.

Метод гілок і меж. Це загальний метод, застосовний як до лінійних, так і до нелінійних задач дискретного (зокрема, цілочислового) програмування. Це комбінаторний метод, реалізований як спрямований перебір варіантів розв'язків оптимізаційних задач зазначеного типу.

Ідея його така. Обчислюють якусь нижню (для задачі мінімізації) чи верхню (для задачі максимізації) оцінку цільової функції $F(x)$ на допустимій множині розв'язків D (наприклад, її нижню чи верхню межу), причому спосіб обчислення оцінки для кожної задачі добирають окремо.

Множину D певним способом розбивають на дві неперетинні підмножини, на кожній з яких переобчислюють оцінки цільової функції $F(x)$. Підмножина з “кращою” оцінкою перспективніша для пошуку, тому її вибирають для подальшого галуження. Інша підмножину вважають кінцевою на цьому етапі. Якщо вона має строго більшу (меншу) оцінку, ніж вибрана для галуження, то надалі її вже не розглядають. Якщо ж відмінність в оцінках для обох підмножин нестрога, то на наступних кроках можливе повернення до якоїсь кінцевої підмножини.

Якщо з кожним новим галуженням оцінка цільової функції “не погіршується”, то отриманий на певному кроці цілочисловий (дискретний) розв'язок — оптимальний розв'язок відповідної початкової задачі, а не то можливе повернення до однієї з попередніх кінцевих підмножин, що має кращу оцінку, ніж отримана на цьому етапі. Тоді процес поділу виконують для цієї підмножини.

Конкретні реалізації методу гілок і меж пов'язані з правилами поділу на підмножини (правилами галуження) та побудови оцінок (меж) значень цільової функції на них. Для задач цілочислового ЛП процедура методу гілок і меж має такий вигляд. Розглянемо частково цілочислову ЗЛП: мінімізувати

$$F(x) = \sum_{j=1}^n c_j x_j$$

для обмежень

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

$$x_j \geq 0, \quad x_j \text{ — цілі, } \quad j = 1, 2, \dots, n, \quad n_1 \leq n,$$

$$0 \leq x_j \leq d_j, \quad j = 1, 2, \dots, n.$$

У загальному випадку деякі зі значень d_j можуть бути нескінченними ($d_j = +\infty$). Уважають, що задана так багатогранна множина обмежена.

Як і в разі розв'язання подібних задач методом відтинання, розглядають допоміжну ЗЛП, отриману з початкової задачі відкиданням умови цілочисловості змінних x_j . Нехай $\mathbf{x}^0 = (x_1^0, x_2^0, \dots, x_n^0)$ — оптимальний розв'язок цієї задачі. Якщо вектор \mathbf{x}^0 цілочисловий, то це оптимальний розв'язок початкової задачі, а не то обчислюють значення цільової функції в точці \mathbf{x}^0 , яке стає її оцінкою (нижньою межею) на початковій допустимій множині розв'язків, і виконують галуження цієї множини.

Вибирають x_k^0 — якусь нецілу компоненту вектора \mathbf{x}^0 . Оскільки в оптимальному розв'язку вона має бути цілою, то накладають додаткові обмеження

$$x_k \leq \left[x_k^0 \right] \quad \text{та} \quad x_k \leq \left[x_k^0 \right] + 1,$$

де $[z]$ — ціла частина числа z .

Отже, початкова множина допустимих розв'язків розпадається на дві підмножини: одна з них містить як додаткове обмеження першу нерівність, а інша — другу. Графічно це має вигляд вилучення одичної смуги з початкової множини (рис. 3.2).

На кожній із новоутворених підмножин розв'язують задачу мінімізації допоміжної нецілочислової ЗЛП. Значення цільової функції на цих підмножинах вибирають як її оцінки (нижні межі). Якщо якомусь із розв'язків відповідає менше значення цільової функції й до того ж він цілочисловий, то це оптимальний розв'язок початкової частково цілочислової задачі, а не то підмножину з нижчою оцінкою цільової функції так само розбивають на підмножини відносно однієї з нецілих компонент відповідного розв'язку допоміжної ЗЛП. Якщо оцінки цільової функції, отримані на попередніх етапах для кінцевих

підмножин (не розгалужуваних), кращі, ніж отримані на останньому етапі для обох підмножин, то для галуження вибирають ту з них, що має найменшу оцінку. Обчислювальний процес ітеративно повторюють починаючи з цієї підмножини.

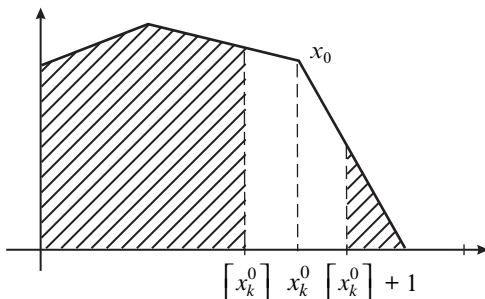


Рис. 3.2

Зауважимо, що для цілком цілочислової ЗЛП з цілими коефіцієнтами $c_j, j = 1, 2, \dots, n$, як оцінку цільової функції $F(x)$ можна брати $\lceil F(x) \rceil = \lceil F(x) \rceil + 1$ — найближче ціле число після значення $F(x)$. Наприклад, $\lceil 1 \rceil = 2, \lfloor 3 \rfloor = 2, \lfloor -1 \rfloor = -1$.

Приклад 3.3. Розв'язати задачу

$$F(x) = -x_1 - 3x_2 \rightarrow \min$$

для обмежень

$$\begin{cases} x_1 + 4x_2 \leq 14, \\ 2x_1 + 3x_2 \leq 12, \\ x_1 \geq 0, x_2 \geq 0, x_1, x_2 \text{ — цілі.} \end{cases}$$

Позначимо як D множину, задану обмеженнями задачі.

1. Розв'яжемо допоміжну ЗЛП (без вимоги цілочисловості змінних x_1, x_2):

$$F(x) = -x_1 - 3x_2 \rightarrow \min,$$

$$\begin{cases} x_1 + 4x_2 \leq 14, \\ 2x_1 + 3x_2 \leq 12, \\ x_1 \geq 0, x_2 \geq 0. \end{cases} \quad (3.4)$$

Розв'язок цієї задачі $x^0 = (6/5; 16/5)$ не цілий (рис. 3.3). Обчислимо нижню межу цільової функції на множині D . Візьмемо $z(D) =]F(x^0)[=]-54/5[=]-54/5[+ 1 = -11 + 1 = -10$.

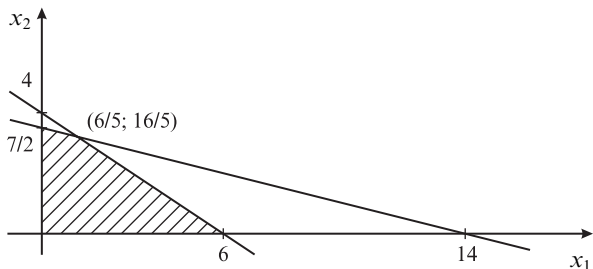


Рис. 3.3

Проведемо галуження множини $D^0 = D = D^{1,1} \cup D^{1,2}$, де $D^{1,1}$ крім обмежень (3.4) містить обмеження

$$x_1 \leq [6/5],$$

а $D^{1,2}$ відрізняється від попередньої обмеженням

$$x_1 \leq [6/5] + 1.$$

2. На обох підмножинах розв'яжемо задачу мінімізації цільової функції (рис. 3.4).

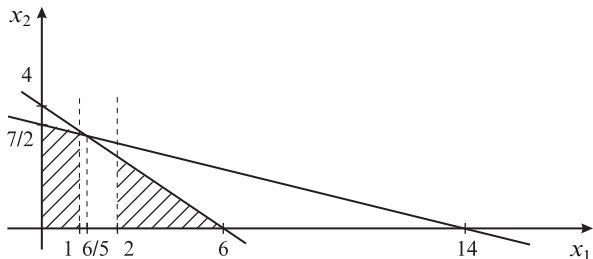


Рис. 3.4

Отримаємо розв'язки $\mathbf{x}^{1,1} = (1; 13/4)$ та $\mathbf{x}^{1,2} = (2; 8/3)$; відповідні значення цільової функції $F(\mathbf{x}^{1,1}) = -43/4$ та $F(\mathbf{x}^{1,2}) = -10$. Тоді оцінки цільової функції — $z(D^{1,1}) =]-43/4[= -10$ і $z(D^{1,2}) =]-10[= -10$. Оскільки посилені оцінки на обох підмножинах однакові, то розглянемо тільки значення цільової функції. За цією оцінкою “кандидат” на галуження — множина $D^{1,1}$.

Виконаємо галуження множини $D^{1,1} = D^{2,1} \cup D^{2,2}$, перша з яких містить додаткове обмеження

$$x_2 \leq [13/4],$$

а друга — обмеження

$$x_2 \leq [13/4] + 1.$$

Позаяк останнє лежить за межами допустимої множини, то множина $D^{2,2}$ порожня. Візьмемо $z(D^{2,2}) = +\infty$.

Розв'язок допоміжної ЗЛП на множині $D^{2,1}$ має вигляд $\mathbf{x}^{2,1} = (1; 3)$; $F(\mathbf{x}^{2,1}) = -10$ (рис. 3.5).

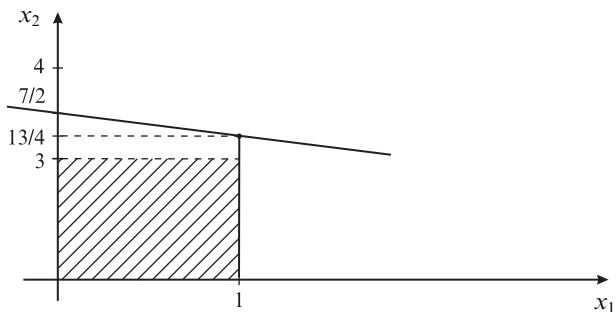


Рис. 3.5

Посилена оцінка цільової функції на цій множині не погіршується порівняно з попередніми ($z(D^0)$, $z(D^{1,1})$, $z(D^{1,2})$), тобто отриманий розв'язок відповідає критерію оптимальності. Тому це оптимальний розв'язок початкової цілочислової задачі, тобто

$$\mathbf{x}^* = (1; 3); \quad F(\mathbf{x}^*) = -10.$$

Очевидний недолік методу гілок і меж у разі розв'язання задач із великою розмірністю полягає в тому, що потрібно перебирати багато варіантів допоміжних задач. Однак цю ваду можна перебороти, якщо шукати не оптимальний розв'язок, а який-небудь близький до оптимального. Про ступінь близькості та швидкість збіжності до екстремуму можна зробити висновки за зміною значень оцінок цільової функції: якщо оцінки на наступних ітераціях мало змінюються, то пошук можна припиняти.

На завершення зазначимо, що не слід застосовувати цілочислові моделі без нагальної потреби, особливо коли йдеться про масовий випуск однорідної продукції. Округлення в таких задачах цілком виправдане на відміну від тих, де розглядають дрібносерійне унікальне обладнання, складні технічні комплекси та ін., для яких вимога цілочисловості принципова.

Контрольні запитання та завдання

1. Які задачі називають цілочисловими ЗЛП?
2. Формалізація яких економічних задач дає задачі цілочислового програмування?
3. Як умова цілочисловості впливає на допустиму множину розв'язків ЗЛП?
4. Які методи застосовують для розв'язання задач цілочислового програмування?
5. Сформулюйте головну ідею методів відтинання.
6. Яка роль двоїстого симплекс-методу в розв'язанні задач цілочислового програмування?
7. Яку ідею покладено в основу методу гілок і меж?
8. Яка послідовність обчислень за методом гілок і меж?

4. НЕЛІНІЙНЕ ПРОГРАМУВАННЯ

Лінійні функції не завжди дають змогу адекватно описати досліджуване явище чи процес. Наприклад, у реальних задачах ціна товару залежить від його кількості (масове виробництво забезпечує нижчу ціну товарів порівняно з індивідуальним). У свою чергу, використання ресурсів може мати не пропорційний характер, а змінюватися стрибкоподібно. Для врахування цих і подібних особливостей реального виробництва потрібно застосовувати загальніші та складніші моделі та розв'язувати нелінійні оптимізаційні задачі, які й складають розділ нелінійного програмування.

Розглянемо декілька прикладів економічних задач, формалізація яких дає задачу нелінійного програмування (ЗНП).

Мінімізація витрат виробництва. Нехай витрати, потрібні для виробництва x одиниць продукту A , задано залежністю

$$K(x) = ax^K, \quad x > 0, \quad a > 0.$$

Якщо $K > 1$, то загальні витрати, починаючи з якогось x_0 , збільшуються швидше, ніж кількість вироблених одиниць продукту A (*прогресія витрат*). Очевидно, що це економічно небажана ситуація.

Якщо ж $0 < K < 1$, то загальні витрати збільшуються повільніше, ніж кількість вироблених одиниць продукту A (*регресія витрат*). Така ситуація (ефект крупносерійного виробництва) буває найчастіше.

Нарешті, в окремих випадках регресії витрат правдиве співвідношення

$$K(x) = a - bx^K, \quad x > 0, \quad a, b, K > 0.$$

Якщо в загальному випадку витрати на виробництво x_j одиниць продукту A_j ($j = 1, 2, \dots, n$) на якомусь підприємстві задано у вигляді

$$K(x_1, x_2, \dots, x_n) = \sum_{j=1}^n a_j x_j^{K_j}, \quad x_j > 0, \quad j = 1, 2, \dots, n,$$

де $a_j, K_j > 0$, а кількість кожного виду виробленої продукції A_j має бути не меншою ніж $b_j (j = 1, 2, \dots, n)$ і задовольняти умові на асортимент $\sum_{j=1}^n d_j x_j \geq d$, де $d_j, d > 0$, то мінімізація витрат на підприємстві означає, що потрібно знайти мінімум нелінійної функції $K(x)$ на мно-

$$\text{жині } G = \{(x_1, x_2, \dots, x_n): \sum_{j=1}^n d_j x_j \geq d, x_j \geq b_j, j = 1, 2, \dots, n\}.$$

Оптимальний вибір факторів на основі технологічної виробничої функції. Нехай випуск продукції підприємства y (у вартісному чи натуральному виразі) залежить від виробничих факторів x_1, x_2, \dots, x_n (наприклад, витрат праці, капітальних вкладень та ін.). Виявлено, що технологічна виробнича функція має вигляд $y = a x_1^{\alpha_1} x_2^{\alpha_2} \dots x_n^{\alpha_n}$, де $a, \alpha_1, \alpha_2, \dots, \alpha_n$ — відомі числові параметри, які визначаються на основі статистичних даних. Нехай витрати, пов'язані з використанням одиниці кожного j -го фактора, дорівнюють $c_j (j = 1, 2, \dots, n)$. Потрібно досягти максимально можливого обсягу випуску продукції за умови, що загальна вартість факторів не перевищує c .

Математична модель задачі має такий вигляд: знайти такі значення змінних x_1, x_2, \dots, x_n , щоб одержати максимально можливий обсяг продукції:

$$a x_1^{\alpha_1} x_2^{\alpha_2} \dots x_n^{\alpha_n} \rightarrow \max$$

за умови, що виконуються обмеження на загальні витрати на всі виробничі фактори

$$\sum_{j=1}^n c_j x_j \leq c$$

й обсяг кожного фактора не може бути від'ємним, тобто

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

У загальному вигляді ці й подібні задачі можна сформулювати в термінах оптимізації, а саме: знайти максимальне чи мінімальне значення функції $f(\mathbf{x})$ на заданій множині D розв'язків $\mathbf{x} = (x_1, x_2, \dots, x_n)$.

Для простоти викладу припустимо, що множину D задано скінченною системою рівнянь і нерівностей (хоча існують також інші способи задання цієї множини).

Отже, розглянемо таку задачу: нехай задано функції $f(\mathbf{x})$, $\varphi_i(\mathbf{x})$, $i = 1, 2, \dots, m$, $\psi_j(\mathbf{x})$, $j = 1, 2, \dots, p$, що залежать від багатьох змінних, тобто $\mathbf{x} = (x_1, x_2, \dots, x_n)$. Потрібно знайти $\min f(\mathbf{x})$ для обмежень $\varphi_i(\mathbf{x}) \leq 0$, $i = 1, 2, \dots, m$; $\psi_j(\mathbf{x}) = 0$, $j = 1, 2, \dots, p$.

Якщо всі функції f , φ_i , ψ_j лінійні, то відповідна задача являє собою ЗЛП. Якщо хоча б одна з функцій нелінійна, то маємо ЗНП. Зокрема, у наведених прикладах цільові функції $K(x)$ та $y = f(x_1, x_2, \dots, x_n)$ нелінійні, обмеження-нерівності в обох випадках лінійні, а обмежень-рівностей немає.

Усі обмеження, що задають ЗНП, у разі потреби можна звести до одного вигляду. Це може бути система, що складається лише з нерівностей або рівнянь. У першому випадку всі рівняння $\psi_j(\mathbf{x}) = 0$ можна замінити двома нерівностями $\psi_j(\mathbf{x}) \geq 0$ та $-\psi_j(\mathbf{x}) \geq 0$, а в другому — додати фіктивні змінні s_i та від нерівностей $\varphi_i(\mathbf{x}) \geq 0$ перейти до рівнянь $\varphi_i(\mathbf{x}) - s_i^2 = 0$, $i = 1, 2, \dots, m$.

Процес розв'язання ЗНП значно ускладнюють такі їх властивості:

1) множина допустимих планів (розв'язків) D може мати досить складну структуру (наприклад, бути неопуклою чи незв'язною);

2) глобальний мінімум (максимум) цільової функції може досягатися за межами множини D , а якщо він досягається на межі D , то він може не збігатись із жодним із локальних мінімумів (максимумів);

3) цільова функція може бути недиференційовною.

Через названі властивості для кожної ЗНП потрібні особливі методи дослідження, і загального методу розв'язання немає.

Виділяють декілька класів найрозповсюдженіших ЗНП.

1. *Класичні задачі оптимізації* полягають у визначенні мінімуму (максимуму) функції $f(\mathbf{x})$, де $\mathbf{x} = (x_1, x_2, \dots, x_n)$, для обмежень-рівнянь $g_i(\mathbf{x}) = 0$, $i = 1, 2, \dots, m$, $m < n$. Якщо $m = 0$, маємо задачу без обмежень. Тоді істотна вимога гладкості (наявність у функцій f і g_i неперервних частинних похідних щонайменше до другого порядку включно). Класичні задачі оптимізації можна розв'язувати (хоча б у принципі) класичними методами з використанням апарату диференціального

числення. Однак для розв'язання багатьох практичних задач ці методи не завжди прийнятні через громіздкі обчислення.

2. *Задачі квадратичного програмування.* У них цільова функція квадратична:

$$f(\mathbf{x}) = \sum_{j=1}^n c_j x_j + \sum_{k=1}^n \sum_{j=1}^n d_{kj} x_k x_j,$$

а обмеження лінійні:

$$\varphi_i(\mathbf{x}) = \sum_{j=1}^n a_{ij} x_j - b_i \leq 0, \quad i = 1, 2, \dots, m.$$

3. *Задачі опуклого програмування.* У них цільова функція $f(\mathbf{x})$ і функції обмежень $g_i(\mathbf{x})$, $i = 1, 2, \dots, m$, опуклі (увігнуті).

Як було зазначено раніше, симплекс-метод розв'язання ЗЛП дає точний розв'язок задачі за скінченну кількість кроків. До нелінійних задач зазвичай застосовують наближені методи розв'язання, які дають змогу знайти певний розв'язок оптимізаційної задачі з певною точністю. Однотипні (ітераційні) обчислення повторюють доти, доки не буде досягнуто потрібної точності.

Геометрична інтерпретація задачі нелінійного програмування

Для геометричного розв'язання задачі математичного програмування істотне поняття лінії рівня цільової функції.

Означення 4.1. Множина точок $\mathbf{x} = (x_1, x_2, \dots, x_n)$ називається **лінією рівня функції** $f(\mathbf{x})$, якщо в усіх цих точках значення функції не змінюються, тобто $f(\mathbf{x}) = \text{const}$.

Звісно, різним константам відповідають різні лінії рівня, однак усі вони мають спільну форму. Наприклад, для лінійної функції $f(\mathbf{x}) = ax_1 + bx_2$ лінії рівня — це паралельні прямі $ax_1 + bx_2 = \text{const}$, де значення сталої визначає зсув прямої відносно початку координат. Для квадратичної функції $f(\mathbf{x}) = x_1^2 + x_2^2$ лінії рівня — концентричні кола з різними радіусами $r = \sqrt{\text{const}}$ і центром у початку координат. Для функції $f(\mathbf{x}) = (x_1 - a)^2 + (x_2 - b)^2$ лінії рівня — також концентричні кола, але з центром у точці (a, b) , а якщо розглядати функцію

$f(\mathbf{x}) = c(x_1 - a)^2 + d(x_2 - b)^2$, де $c > 0$, $d > 0$, то для неї лінії рівня — концентричні еліпси, що мають центр у точці (a, b) .

Приклад 4.1. Нехай цільова функція задачі залежить від двох змінних $(\mathbf{x} = (x_1, x_2))$ та має вигляд $f(\mathbf{x}) = x_1/2 + 2x_2$, а область допустимих розв'язків задано обмеженнями

$$\begin{cases} x_1 + x_2 \leq 6, \\ x_1 - x_2 \leq 1, \\ x_1 - 2x_2 \geq -8, \\ x_1 \geq 0, x_2 \geq 0. \end{cases}$$

Потрібно знайти найбільше значення цільової функції на множині допустимих розв'язків. Це очевидно, ЗЛП, і її можна розв'язати графічно (рис. 4.1).

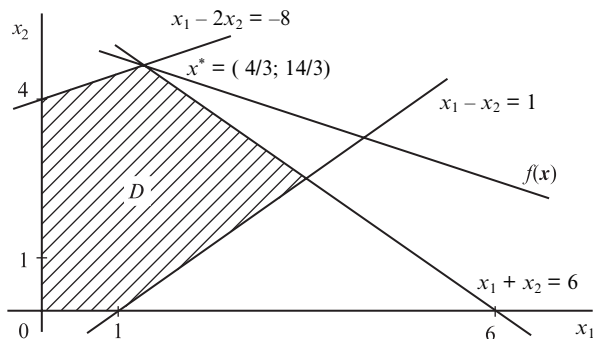


Рис. 4.1

Як видно з графіка, розв'язок задачі лежить у вершині допустимої області й дорівнює $\mathbf{x}^* = (4/3; 14/3)$; $f(\mathbf{x}^*) = 10$.

Приклад 4.2. Якщо для тих самих обмежень задати нелінійну цільову функцію

$$f(\mathbf{x}) = 10(x_1 - 3,5)^2 + 20(x_2 - 4)^2,$$

то отримаємо ЗНП з нелінійною (квадратичною) функцією цілі та лінійними обмеженнями. Розв'язання цієї задачі, як видно з рис. 4.2,

зводиться до визначення x_1^* , x_2^* і $z^* = f(x_1^* + x_2^*)$, що задовольняють системі рівнянь

$$\begin{cases} x_1^* + x_2^* = 6, \\ f(\mathbf{x}^*) = z^* = 10(x_1^* - 3,5)^2 + 20(x_2^* - 4)^2, \\ x_2^* - 4 = 0,5(x_1^* - 3,5). \end{cases}$$

Її отримано з очевидних із геометричного погляду фактів:
 оптимальний розв'язок — точка (x_1^*, x_2^*) , що лежить на прямій $x_1 + x_2 = 6$ і лінії рівня $f(\mathbf{x}^*) = z^*$;
 у точці (x_1^*, x_2^*) пряма $x_1 + x_2 = 6$ дотикається до лінії рівня $f(\mathbf{x}^*) = z^*$,
 звідки

$$-1 = \frac{dx_2}{dx_1} = -\frac{\partial z / \partial x_1}{\partial z / \partial x_2} = -\frac{20(x_1 - 3,5)}{40(x_2 - 4)} \Big|_{x_1=x_1^*, x_2=x_2^*}.$$

Розв'язавши наведену вище систему рівнянь, отримаємо $\mathbf{x}^* = (2,5; 3,5)$;
 $z^* = 15$.

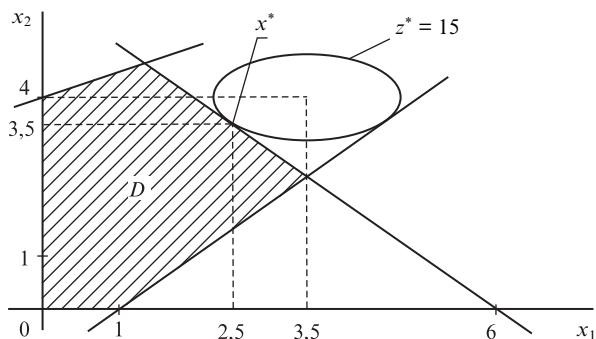


Рис. 4.2

Зауважимо, що оптимальне значення цільової функції досягається не у вершині допустимої області, як у ЗЛП, а в точці, що належить одній зі сторін допустимої області.

Класичний метод дослідження нелінійних оптимізаційних задач

Вибір методів розв'язання ЗНП залежить від таких властивостей функцій, як обмеженість, неперервність, диференційовність, опуклість та ін. Відомо, що неперервна функція на замкненій обмеженій множині принаймні один раз набуває мінімального та максимального значень. При цьому точки екстремуму являють собою або стаціонарні точки, або точки межі, або такі, де функція недиференційовна.

Нагадаємо, що для функції $f(\mathbf{x})$ стаціонарні точки — це ті, у яких її похідна дорівнює нулю, тобто $f'(\mathbf{x}) = 0$. Для функції багатьох змінних $f(\mathbf{x}) = f(x_1, x_2, \dots, x_n)$ перша похідна — це вектор, що складається з перших частинних похідних за всіма змінними:

$$\nabla f(\mathbf{x}) = \left(\frac{\partial f(\mathbf{x})}{\partial x_1}, \frac{\partial f(\mathbf{x})}{\partial x_2}, \dots, \frac{\partial f(\mathbf{x})}{\partial x_n} \right).$$

Його називають *градієнтом функції*. Тоді умова $f'(\mathbf{x}) = 0$ означає, що стаціонарна точка — це така точка $\mathbf{x} = (x_1, x_2, \dots, x_n)$, у якій виконується система рівнянь

$$\left\{ \begin{array}{l} \frac{\partial f(\mathbf{x})}{\partial x_1} = 0, \\ \frac{\partial f(\mathbf{x})}{\partial x_2} = 0, \\ \dots\dots\dots \\ \frac{\partial f(\mathbf{x})}{\partial x_n} = 0. \end{array} \right.$$

Щоб визначити, чи стаціонарні точки являють собою точки екстремуму, потрібно дослідити в них другу похідну. Для функції багатьох змінних $f(\mathbf{x}) = f(x_1, x_2, \dots, x_n)$ похідні другого порядку записують у вигляді *матриці Гессе*:

Якщо визначники мають однаковий знак (а саме, додатні), то \mathbf{x}^0 — точка відносного (локального) мінімуму. Якщо ж $\Delta_1, \Delta_2, \dots, \Delta_n$ по черзі змінюють знак, починаючи з мінуса ($\Delta_1 < 0, \Delta_2 > 0, \Delta_3 < 0$ і т. д.), то \mathbf{x}^0 — точка відносного максимуму. Поки дослідження проводять в окремих точках, мова йде саме про *локальні (місцеві) екстремуми*. Проаналізувавши всю допустиму область розв'язків, можна виділити серед локальних екстремумів найбільший і найменший, які називають *глобальними*.

Приклад 4.3. Нехай функцію $f(\mathbf{x}) = f(x_1, x_2) = 10x_1 + 20x_2 + x_1x_2 - 2x_1^2 - 2x_2^2$ задано на всьому просторі R^2 . Для обчислення відносного екстремуму цієї функції маємо два рівняння:

$$\frac{\partial f}{\partial x_1} = 10 + x_2 - 4x_1 = 0, \quad \frac{\partial f}{\partial x_2} = 20 + x_1 - 4x_2 = 0,$$

розв'язавши які, знайдемо $x_1^0 = 4, x_2^0 = 6$. Матриця Гессе для цієї функції має вигляд

$$H = \begin{pmatrix} -4 & 1 \\ 1 & -4 \end{pmatrix}, \quad \text{і} \quad \Delta_1 = -4 < 0, \quad \Delta_2 = \begin{vmatrix} -4 & 1 \\ 1 & -4 \end{vmatrix} = 15 > 0.$$

Отже, $\mathbf{x}^0 = (4; 6)$ — точка відносного максимуму. Оскільки функція $f(\mathbf{x})$ не має в області визначення інших стаціонарних точок, то \mathbf{x}^0 — також точка глобального максимуму.

Для задачі з обмеженнями класичний метод пошуку умовного екстремуму можна описати так.

1. Знаходять множину $S_1(\mathbf{x})$ усіх стаціонарних точок функції $f(\mathbf{x})$ усередині допустимої множини D . Кожну з них досліджують на мінімум (максимум) і визначають серед них ті, де функція $f(\mathbf{x})$ набуває найменшого (найбільшого) значення. Це точки глобального екстремуму.

2. Досліджують точки межі $S_2(\mathbf{x})$ і знаходять ті з них, де функція $f(\mathbf{x})$ досягає мінімуму (максимуму). Для цього вибирають довільну межу, наприклад $g_1(\mathbf{x}) = 0$. Якщо це рівняння можна розв'язати відносно компонент точки $\mathbf{x} = (x_1, x_2, \dots, x_n)$, то, підставивши їх у вираз для цільової функції, отримуємо *задачу безумовної оптимізації*, тобто задачу без обмежень. Такий прямолінійний підхід потребує

значних обчислювальних витрат, і його застосовують лише в найпростіших випадках, у разі невеликої кількості обмежень $g_i(\mathbf{x}) = 0$, $i = 1, 2, \dots, m$, до того ж таких, що кожне рівняння $g_i(\mathbf{x}) = 0$ можна легко розв'язати щодо всіх змінних x_1, x_2, \dots, x_n , тобто можна одержати явні функції у вигляді

$$x_k = \varphi_k(x_1, \dots, x_{k-1}, x_{k+1}, x_n), \quad k = 1, 2, \dots, n.$$

Зокрема, це можливо, якщо функції $g_i(\mathbf{x})$ лінійні. В інших ситуаціях для обчислення точок екстремуму функції $f(\mathbf{x})$ на межі допустимої множини застосовують інші методи.

Метод Лагранжа для задач умовної оптимізації

Задача пошуку екстремуму функції, що має обмеження на змінні, називається *задачею умовної оптимізації*. Якщо на змінні не накладено обмежень, то маємо задачу безумовної оптимізації.

Метод Лагранжа застосовують для розв'язання ЗНП такого вигляду: знайти

$$f(\mathbf{x}) \rightarrow \min \tag{4.1}$$

в обмеженому просторі

$$d = \{x: g_i(x) = 0, i = 1, 2, \dots, m\}. \tag{4.2}$$

Якщо функції $f(\mathbf{x})$, $g_i(\mathbf{x})$ неперервно диференційовні, то для задачі (4.1), (4.2) будують *функцію Лагранжа*

$$L(\mathbf{x}, \lambda) = f(\mathbf{x}) + \sum_{i=1}^m \lambda_i g_i(\mathbf{x}).$$

Нехай у якійсь точці $\mathbf{x}^0 = (x_1^0, x_2^0, \dots, x_n^0)$ існує такий вектор $\lambda^0 = (\lambda_1^0, \lambda_2^0, \dots, \lambda_n^0)$, що виконується необхідна умова екстремуму функції $L(\mathbf{x}, \lambda)$

$$\frac{\partial L(\mathbf{x}^0, \lambda^0)}{\partial x_j} = 0, \quad j = 1, 2, \dots, n; \quad \frac{\partial L(\mathbf{x}^0, \lambda^0)}{\partial x_i} = 0, \quad i = 1, 2, \dots, m.$$

Тоді \mathbf{x}^0 — розв'язок задачі (4.1) з обмеженнями (4.2).

Приклад 4.4. Є два способи виробництва якогось продукту. Нехай x_1, x_2 — кількості продукту, виробленого відповідним способом. Витрати виробництва залежать від кількості виробленого продукту x_1 та x_2 і їх задано співвідношеннями

$$H_1(x_1) = a_0 + a_1x_1 + a_2x_1^2, \quad a_0, a_1, a_2 > 0;$$

$$H_2(x_2) = b_0 + b_1x_2 + b_2x_2^2, \quad b_0, b_1, b_2 > 0.$$

За якийсь проміжок часу потрібно виробити c одиниць продукції (тобто $c = x_1 + x_2$), застосувавши обидва способи виробництва так, щоб мінімізувати загальні витрати.

Математична модель цієї задачі така: знайти

$$f(\mathbf{x}) = H_1(x_1) + H_2(x_2) \rightarrow \min$$

за умови

$$g_i(\mathbf{x}) = c - x_1 - x_2 = 0.$$

При цьому функція Лагранжа згідно із загальним правилом її побудови набирає вигляду

$$L(x_1, x_2, \lambda) = a_0 + a_1x_1 + a_2x_1^2 + b_0 + b_1x_2 + b_2x_2^2 + \lambda(c - x_1 - x_2),$$

звідки

$$\begin{cases} \frac{\partial L}{\partial x_1} = a_1 + 2a_2x_1 - \lambda = 0, \\ \frac{\partial L}{\partial x_2} = b_1 + 2b_2x_2 - \lambda = 0, \\ \frac{\partial L}{\partial \lambda} = c - x_1 - x_2 = 0. \end{cases}$$

Розв'язавши цю систему, знаходимо невідомі:

$$x_1^0 = \frac{b_2}{a_2 + b_2} c + \frac{b_1 - a_1}{2(a_2 + b_2)}; \quad x_2^0 = \frac{a_2}{a_2 + b_2} c - \frac{b_1 - a_1}{2(a_2 + b_2)}.$$

Практичне значення методу Лагранжа полягає в тому, що він дає змогу від задачі умовної оптимізації перейти до задачі безумовної оптимізації, поєднавши в одній функції й цільову функцію початкової задачі, і її обмеження. Однак розв'язання системи рівнянь у разі великої кількості змінних і обмежень на них — також досить складна задача. Тому подібний підхід застосовний до неширокого кола задач, для яких система рівнянь лінійна чи може бути зведена до лінійної. Цей метод застосовують тоді, коли потрібно знайти аналітичний (у вигляді формули) розв'язок оптимізаційної задачі (наприклад, у певних теоретичних дослідженнях).

Методи розв'язання задач безумовної оптимізації

Для практичних задач зазвичай становить інтерес тільки кількісний розв'язок, тому для його пошуку використовують певні ітераційні процедури, які дають змогу за скінченну кількість кроків визначити точку локального екстремуму цільової функції чи наблизитися до нього. Однак у разі застосування цих процедур (так званих *прямих методів розв'язання*) дуже важливі питання їх збіжності: перш ніж застосовувати будь-який ітераційний метод пошуку екстремуму функції, потрібно бути впевненим, що за скінченну кількість кроків за цими методами можна наблизитися до точки екстремуму — потрапити в досить малий її окіл. Тому й у цьому разі задачі поділяють на певні класи та розробляють для кожного свої методи пошуку оптимуму цільової функції.

Серед нелінійних оптимізаційних задач насамперед виділяють задачі безумовної оптимізації (без обмежень на змінні) і задачі опуклого програмування.

Залежно від властивостей цільової функції (неперервність, диференційовність) обирають метод ітераційного пошуку.

Якщо цільова функція диференційовна, то для пошуку її стаціонарних точок найчастіше застосовують градієнтний метод. Його ідея полягає в тому, що градієнт функції, обчислений у довільній точці,

показує напрямок найбільшого зростання функції в її околі. Тому якщо з якоїсь точки перейти до іншої цьому напрямку, то значення цільової функції може збільшитися (за умови, що нова точка лежить у якомусь околі попередньої). Градієнт, обчислений у новій точці, показує подальший напрямок зростання. Пересуваючись від точки до точки, можна потрапити в окіл локального максимуму цільової функції. Очевидно, що, вибираючи як напрямок руху протилежний до показуваного вектором-градієнтом, тобто напрямком антиградієнта, так само можна потрапити в точку локального мінімуму. Щоб прискорити збіжність градієнтного методу, потрібно якомога “краще” визначити відстань між точками — *крок*. Залежно від способу його визначення розглядають різні варіанти цього методу, найпопулярніші з яких — методи найшвидшого спуску та подрібнення кроку.

Метод найшвидшого спуску. Якщо, переходячи від однієї точки до іншої, зробити занадто великий крок, тобто вийти за межі досить малого околу попередньої точки, то значення цільової функції може виявитися “гіршим”, ніж попереднє. Якщо зробити занадто малий крок, то наближення до локального екстремуму може бути повільним. Тому для зменшення загальної кількості кроків на кожній ітерації методу розв’язують задачу визначення оптимального кроку — *одновимірної оптимізації*. Ознака досягнення стаціонарної точки — досить мала відмінність між значеннями координат послідовних точок (досить мала відстань між ними) і водночас досить мала норма (довжина) вектора-градієнта. Останнє можливо, якщо всі координати вектора-градієнта близькі до нуля.

Метод подрібнення кроку. Задача одновимірної оптимізації, розв’язувана на кожній ітерації методу найшвидшого спуску, може, у свою чергу, потребувати значних обчислювальних витрат, а це значно зменшує ефективність методу загалом. Тому інколи для обчислення кроку застосовують такий підхід. Вибирають якийсь початковий крок, наприклад $\lambda = 1$, і перевіряють значення цільової функції в новій точці, отриманій унаслідок зміщення в напрямку градієнта (антиградієнта) з попередньої точки на відстань $\lambda = 1$. Якщо це значення погіршилося порівняно зі значенням у попередній точці, то крок удвічі зменшують, а не то удвічі збільшують. Процес продовжують доти, доки не буде знайдено потрібну відстань між точками.

Якщо зважити на те, що далеко від стаціонарної точки крок ітерації може бути досить великим, а з наближенням до неї має значно зменшуватися, то його можна вибирати автоматично: спершу вибрати

якесь початкове значення кроку, наприклад узяти $\lambda = 1$, і послідовно зменшувати його від ітерації до ітерації за певним законом (наприклад, законом геометричної прогресії); тоді за певних умов через скінченну кількість кроків також можна наблизитися до стаціонарної точки цільової функції. Критерій завершення обчислень за цим методом може бути таким, як і в методі найшвидшого спуску.

Спільний недолік усіх варіантів градієнтного методу полягає в тому, що вони дають змогу відшукати тільки стаціонарні точки, тобто “підозрілі” на локальний екстремум, до того ж усі знайдені розв’язки суттєво залежать від точки, з якої почато обчислення — *початкового наближення*. Для пошуку глобальних екстремумів потрібно досліджувати всі стаціонарні точки; їх можна визначати завдяки правильному підбору початкових наближень, що, у свою чергу, потребує додаткових досліджень задачі.

Однак є клас задач, для яких градієнтні методи завжди дають змогу визначити глобальний екстремум, незважаючи на початкове наближення. Це задачі опуклого програмування.

Елементи опуклого аналізу

Задачі, у яких цільова функція опукла (увігнута) і допустима множина розв’язків опукла, називаються *задачами опуклого програмування (ЗОП)*. Їх перевага полягає в тому, що завжди існує мінімум (максимум) цільової функції, і він абсолютний (глобальний).

Означення 4.2. Множина X називається *опуклою*, якщо будь-які дві її точки можна з’єднати відрізком, що цілком належить цій множині, або, що те саме, для будь-яких його точок x^1 , x^2 та довільного числа λ : $0 < \lambda < 1$ точка $y = \lambda x^1 + (1 - \lambda)x^2$ також належить цій множині.

Приклади опуклих множин — відрізок, промінь, пряма, півплощина, куля.

Означення 4.3. Функція $f(x)$, визначена на якійсь опуклій множині X , називається *опуклою*, якщо для будь-яких точок x^1 і x^2 з множини X і довільного числа λ : $0 < \lambda < 1$ виконується нерівність

$$f(\lambda x^1 + (1 - \lambda)x^2) \leq \lambda f(x^1) + (1 - \lambda) f(x^2).$$

Означення 4.4. Функція $f(x)$, визначена на опуклій множині X , називається *увігнутою*, якщо для будь-яких точок x^1 та x^2 з множини X і довільного числа λ : $0 < \lambda < 1$ виконується нерівність

$$f(\lambda x^1 + (1 - \lambda)x^2) \geq \lambda f(x^1) + (1 - \lambda)f(x^2).$$

Зокрема, для функції однієї змінної ця нерівність означає, що відрізок, який з'єднує довільну пару точок її графіка, ні в яких інших точках не перетинає цей графік, й увесь графік опуклої функції лежить нижче відрізка, що з'єднує точки $(x^1; f(x^1))$ та $(x^2; f(x^2))$, а графік увігнутої функції лежить вище цього відрізка (рис. 4.3).

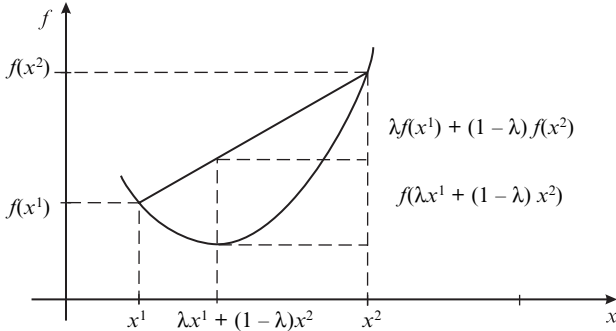


Рис. 4.3

Якщо функція $f(x)$ опукла, то функція $-f(x)$ увігнута. Якщо функція $g(x)$ опукла, то множина W , задана нерівністю $g(x) \leq 0$, опукла. Перетин опуклих множин, тобто множина, задана системою обмежень $W = \{x: g_i(x) \leq 0, i = 1, 2, \dots, m\}$, де всі функції $g_i(x)$ опуклі, — чи порожня, чи опукла та замкнена множина.

Задача опуклого програмування має таку загальну постановку: знайти

$$\min f(x) \tag{4.3}$$

для обмежень

$$g_i(x) \leq 0, \quad i = 1, 2, \dots, m, \tag{4.4}$$

де $f(x)$ і $g_i(x)$, $i = 1, 2, \dots, m$ — опуклі функції багатьох змінних $x = (x_1, x_2, \dots, x_n)$.

Якщо функції $f(\mathbf{x})$ і $g_i(\mathbf{x})$, $i = 1, 2, \dots, m$, увігнуті, то задача максимізації

$$\max f(\mathbf{x})$$

для обмежень

$$g_i(\mathbf{x}) \geq 0, \quad i = 1, 2, \dots, m,$$

називається *задачею увігнутого програмування (ЗУП)*.

Можна довести, що обидві ці задачі еквівалентні.

Достатня умова опуклості функції багатьох змінних $f(\mathbf{x})$ — додатна визначеність матриці її других похідних (матриці Гессе) в усіх точках множини D . Для увігнутої функції достатня умова — від'ємна визначеність матриці Гессе.

Якщо множина допустимих розв'язків ЗОП (4.3), (4.4) порожня, то задача не має розв'язків, а якщо вона необмежена, то не існує оптимального розв'язку. На замкненій обмеженій множині ЗОП має таку дуже важливу властивість, яка полегшує пошук розв'язку.

Теорема 4.1. Якщо функція $f(\mathbf{x})$ опукла (увігнута) на множині R^n (у всьому n -вимірному просторі) та в точці \mathbf{x}^* її градієнт дорівнює нулю, то \mathbf{x}^* — точка глобального екстремуму.

Інакше кажучи, будь-який локальний мінімум (максимум) ЗОП глобальний.

Якщо задача опуклого програмування не має обмежень (являє собою задачу безумовної оптимізації), то для визначення її екстремуму застосовують градієнтні методи. Обмеження істотно ускладнюють методи розв'язання нелінійних оптимізаційних задач. Якщо задача має обмеження на змінні, то для пошуку її оптимуму застосовують *спеціальні методи*: проєкції градієнта, допустимих напрямків (метод Зойтендейка), штрафних і бар'єрних функцій.

Як і в градієнтних методах, обчислювальні процедури цих методів також включають визначення напрямку руху та кроку на кожній ітерації. Щодо векторів, які визначають напрямок пересування, уведуть два такі основні поняття.

Означення 4.5. Напрямок $\mathbf{s} = (s_1, s_2, \dots, s_n)$ називається *допустимим (можливим)*, якщо для довільної точки $\mathbf{x}^q = (x_1^q, x_2^q, \dots, x_n^q)$, що належить множині D , існує таке $\lambda > 0$, що точка $\mathbf{x}^{q+1} = \mathbf{x}^q + \lambda \mathbf{s}$ також належить множині D .

Це означає, що незначне зміщення в напрямку s не виводить за межі допустимої області D .

Означення 4.6. Напрямок $s = (s_1, s_2, \dots, s_n)$ називається **прогресивним** (**перспективним**), якщо для довільної точки $x^q = (x_1^q, x_2^q, \dots, x_n^q)$, що належить множині D , існує таке $\lambda > 0$, що $f(x^q) < f(x^q + \lambda s)$ для задачі максимізації й $f(x^q) > f(x^q + \lambda s)$ для задачі мінімізації.

Отже, рух у перспективному напрямку поліпшує значення цільової функції.

На підставі цих означень формулюють критерій перевірки оптимальності точки. Точка x^* являє собою оптимальний план задачі умовної оптимізації, якщо жоден з її допустимих напрямків не прогресивний.

Розглянемо детальніше *метод Зойтендейка*.

Метод можливих напрямків. У цьому методі правило вибору наступної точки, до якої потрібно перейти, суттєво залежить від розміщення поточної точки. Розрізняють дві принципово різні ситуації.

1. Точка лежить усередині області D . Очевидно, що для довільної внутрішньої точки будь-який напрямок допустимий за умови, що крок досить малий. Однак серед усіх можливих напрямків бажано вибирати той, у якому значення функції напевне збільшиться. Як зазначалося раніше, це може бути напрямок градієнта функції. Кроковий множник вибирають так, щоб, по-перше, не вийти за межі допустимої області, по-друге — якомога повніше використати цей напрямок, тобто максимально можливо збільшити значення цільової функції. Для цього відносно крокового множника $\lambda > 0$ розв'язують одновимірну задачу оптимізації, що враховує обидва зазначені аспекти. У результаті отримують таке значення λ , що нова точка виявляється всередині області D чи на її межі. У першому випадку виконують обчислення в уже описаній послідовності, а в другому розв'язують допоміжну ЗЛП.

2. Точка лежить на межі допустимої області. Це означає, що в ній одна чи декілька нерівностей із системи обмежень задачі виконуються як строгі рівняння. Обмеження, яке виконується як рівність, називають *активним*. Точка, що лежить на межі області, має менше допустимих напрямків, ніж внутрішні точки області, бо частина напрямків виводить за допустимі межі. Щоб серед можливих напрямків знайти прогресивний, потрібно розв'язати ЗЛП, яка враховує актив-

ні обмеження початкової задачі та гарантує зростання значення цільової функції. Перехід до лінійної задачі можливий тому, що зазначені умови формулюють у термінах градієнтів і їх скалярних добутоків з векторами-напрямами, тобто система обмежень складається з лінійних функцій, а цільова функція такої ЗЛП — якийсь параметр, найменше значення якого забезпечує максимальне прискорення процесу оптимізації цільової функції нелінійної задачі.

Після того, як прогресивний напрямок знайдено, визначають кроковий множник так, як зазначено в попередньому пункті. Обчислення закінчуються в тому разі, коли досягнуто якоїсь точки x^* , вектор-градієнт котрої майже дорівнює нульовому вектору, тобто має довжину, меншу за досить малу величину $\varepsilon > 0$, що визначає точність обчислень.

Інші методи розв'язання задач опуклого програмування. Потреба враховувати обмеження в ЗНП суттєво ускладнює методи їх розв'язання. Застосовуючи метод можливих напрямків для оптимізації цільової функції, потрібно розв'язувати допоміжну ЗЛП. У разі застосування градієнтних методів до розв'язання ЗНП також часто виникають ситуації, коли напрямок градієнта (антиградієнта) виводить за межі допустимої області. Тоді потрібна операція проектування градієнта на допустиму область, а це досить складна задача, особливо в разі нелінійних обмежень.

Водночас задачі без обмежень розв'язувати порівняно просто, тому задачі умовної оптимізації (з обмеженнями) намагаються звести до безумовних задач. При цьому обмеження початкової задачі вводять до складу цільової функції (наприклад, за допомогою спеціальних множників, як у методі Лагранжа, чи за допомогою штрафних або бар'єрних функцій).

Означення 4.7. Функцію $P(g_1(x), g_2(x), \dots, g_m(x), r)$, де $g_i(x)$, $i = 1, 2, \dots, m$ — функції обмежень умовної задачі оптимізації, r — параметр функції $P(\cdot)$ (ваговий коефіцієнт), називають **штрафною**, якщо зі зростанням значення r ця функція дорівнює нулю, $g_i(x) \leq 0$, $i = 1, 2, \dots, m$, і нескінченності, якщо ця умова не виконується.

Таке означення штрафної функції дає змогу від початкової задачі мінімізації (4.3), (4.4) перейти до послідовності задач безумовної оптимізації: знайти мінімум функції

$$F(x, r) = f(x) + P(g_1(x), g_2(x), \dots, g_m(x), r).$$

Зазвичай як штрафну вибирають функцію

$$P(g_1(\mathbf{x}), g_2(\mathbf{x}), \dots, g_m(\mathbf{x}), r) = \frac{1}{r} \sum_{i=1}^m P_i(g_i(\mathbf{x})),$$

де $r > 0$ — коефіцієнт штрафу, а функції $P_i(g_i(\mathbf{x}))$ означають так:

$$P_i(g_i(\mathbf{x})) = \max \{0, g_i(\mathbf{x})\} \quad \text{або} \quad P_i(g_i(\mathbf{x})) = [\max \{0, g_i(\mathbf{x})\}]^2.$$

Незалежно від способу означення штрафної функції вона має бути така, щоб порушення якогось обмеження початкової ЗНП стало не вигідним для задачі безумовної оптимізації.

Доведено, що в разі вдалого вибору штрафної функції послідовність задач безумовної оптимізації збігається до розв'язку початкової умовної задачі.

Один із різновидів методу штрафних функцій — *метод бар'єрних функцій*. Відмінність його від попереднього методу полягає в тому, що функцію $P(g_1(\mathbf{x}), g_2(\mathbf{x}), \dots, g_m(\mathbf{x}), r)$ апроксимують (наближають) бар'єрними функціями $B(g_1(\mathbf{x}), g_2(\mathbf{x}), \dots, g_m(\mathbf{x}), r)$ із середини допустимої області. Тоді в разі перетину меж допустимої області значення цільової функції $F(\mathbf{x}, r)$ також погіршується. Застосування послідовності функцій дає змогу визначити розв'язок початкової ЗНП із заданою точністю.

Двоїстість у нелінійному програмуванні

Описаний раніше метод множників Лагранжа застосовують до розв'язання класичних задач на умовний екстремум, тобто в тому разі, коли обмеження задачі — рівняння. Поширення цього методу на розв'язання задач з обмеженнями-нерівностями потребує певного теоретичного обґрунтування.

Нехай для задачі умовної оптимізації (4.3), (4.4) складено функцію Лагранжа

$$L(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) + \sum_{i=1}^m \lambda_i g_i(\mathbf{x}).$$

Означення 4.8. Пара векторів $L(\mathbf{x}^0, \boldsymbol{\lambda}^0)$ називається *сідловою точкою* функції Лагранжа $L(\mathbf{x}, \boldsymbol{\lambda})$, якщо для довільних векторів $\mathbf{x} = (x_1, x_2, \dots, x_n)$ і $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \dots, \lambda_m)$ виконується співвідношення

$$L(\mathbf{x}, \boldsymbol{\lambda}) \leq L(\mathbf{x}^0, \boldsymbol{\lambda}^0) \leq L(\mathbf{x}^0, \boldsymbol{\lambda}),$$

які називають *нерівностями сідлової точки*.

Наприклад, для функції $L(\mathbf{x}, \boldsymbol{\lambda}) = -x^2 + \lambda^2$ сідлова точка — це $(x^0, \lambda^0) = (0, 0)$, бо $L(x, 0) = -x^2 \leq 0$, $L(0, \lambda) = \lambda^2 \geq 0$ й водночас

$$L(\mathbf{x}, 0) \leq L(0, 0) \leq L(0, \boldsymbol{\lambda}).$$

Теорема Куна — Таккера, яка посідає центральне місце в теорії нелінійного програмування, пов'язує розв'язання ЗНП з наявністю сідлової точки відповідної функції Лагранжа; її можна розглядати як узагальнення теореми двоїстості, сформульованої в ЛП. Множники Лагранжа λ_i , $i = 1, 2, \dots, m$ являють собою допустимі розв'язки двоїстої ЗЛП й до того ж мають певний економічний зміст.

Теорема 4.2. (Куна — Таккера). Нехай множина $W = \{\mathbf{x}: g_i(\mathbf{x}) \leq 0, i = 1, 2, \dots, m\}$ має внутрішні точки, тобто виконується умова *регулярності Слейтера*: існує таке $\mathbf{x} \in W$, що $g_i(\mathbf{x}) < 0$ для всіх $i = 1, 2, \dots, m$. Тоді для того, щоб точка \mathbf{x}^* була оптимальним розв'язком задачі (4.3), (4.4), необхідно й достатньо, щоб існував такий невід'ємний m -вимірний вектор $\boldsymbol{\lambda}^* = (\lambda_1^*, \lambda_2^*, \dots, \lambda_m^*)$, що пара $(\mathbf{x}^*, \boldsymbol{\lambda}^*)$ — сідлова точка функції Лагранжа

$$L(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) + \sum_{i=1}^m \lambda_i g_i(\mathbf{x}), \quad \mathbf{x} \in R^n, \quad \lambda_i \geq 0,$$

тобто виконуються нерівності

$$f(\mathbf{x}) + \sum_{i=1}^m \lambda_i^* g_i(\mathbf{x}) \geq f(\mathbf{x}^*) + \sum_{i=1}^m \lambda_i g_i(\mathbf{x}^*) \geq f(\mathbf{x}^*) + \sum_{i=1}^m \lambda_i g_i(\mathbf{x}^*),$$

$$\lambda_i \geq 0, \quad i = 1, 2, \dots, m.$$

Крім того, виконується правило доповнювальної нежорсткості

$$\sum_{i=1}^m \lambda_i^* g_i(\mathbf{x}^*) = 0.$$

Наявність сідлової точки функції Лагранжа означає, що виконуються такі умови:

$$\frac{\partial L(\mathbf{x}^0, \boldsymbol{\lambda}^0)}{\partial x_j} \geq 0, \quad (4.5)$$

$$\frac{\partial L(\mathbf{x}^0, \boldsymbol{\lambda}^0)}{\partial x_j} x_j^0 = 0, \quad j = 1, 2, \dots, n,$$

$$\frac{\partial L(\mathbf{x}^0, \boldsymbol{\lambda}^0)}{\partial \lambda_i} = g_i(\mathbf{x}^0) \leq 0, \quad (4.6)$$

$$\frac{\partial L(\mathbf{x}^0, \boldsymbol{\lambda}^0)}{\partial \lambda_i} \lambda_i^0 = \lambda_i^0 g_i(\mathbf{x}^0) = 0, \quad i = 1, 2, \dots, m.$$

Для ЗУП нерівності (4.5) і (4.6) мають протилежні знаки.

Значення теореми Куна — Таккера полягає в тому, що вона дає змогу звести умовну задачу оптимізації до пошуку сідлових точок функції Лагранжа, який, у свою чергу, зводиться до максимізації функції Лагранжа відносно \mathbf{x} і мінімізації відносно $\boldsymbol{\lambda}$.

Розглянемо застосування методу Лагранжа до розв'язання задач опуклого (увігнутого) програмування на певному класі задач квадратичного програмування.

Квадратичне програмування

Один із найпростіших і найважливіших класів ЗНП та, зокрема, ЗОП — задачі квадратичного програмування. Вони вирізняються серед інших умовних оптимізаційних задач тим, що цільова функція в них квадратична, а обмеження — лінійні.

Квадратична функція

$$\begin{aligned} f(\mathbf{x}) &= c_1 x_1 + c_2 x_2 + \dots + c_n x_n + d_{11} x_1^2 + d_{12} x_1 x_2 + \dots + d_{ij} x_i x_j + \dots + d_{nn} x_n^2 = \\ &= \sum_{j=1}^n c_j x_j + \sum_{i=1}^n \sum_{j=1}^n d_{ij} x_i x_j = (\mathbf{c}, \mathbf{x}) + \frac{1}{2} (\mathbf{D}\mathbf{x}, \mathbf{x}) \end{aligned}$$

залежно від матриці D (завжди симетричної) може бути опуклою (якщо D невід'ємно визначена) чи увігнутою (якщо D недодатно визначена). У теорії лінійних нерівностей доведено, що система лінійних нерівностей задає опуклу множину. Тому задача мінімізації квадратичної функції $[(Dx, x) + (c, x)] \rightarrow \min$ з невід'ємно визначеною матрицею D та лінійними обмеженнями $Ax \leq b$ — це ЗОП, а задача максимізації квадратичної функції $[(Dx, x) + (c, x)] \rightarrow \max$ для лінійних обмежень $Ax \geq b$, де D — симетрична недодатно визначена матриця, — це ЗУП.

Теорему Куна — Таккера для задачі квадратичного програмування з увігнутою цільовою функцією можна сформулювати так.

Теорема 4.3. Для того, щоб точка $x^0 = (x_1^0, x_2^0, \dots, x_n^0)$ була оптимальним планом задачі квадратичного програмування, необхідно й достатньо, щоб існували невід'ємні m -вимірні вектори $\lambda^0 = (\lambda_1^0, \lambda_2^0, \dots, \lambda_m^0)$, $w = (w_1, w_2, \dots, w_m)$ і невід'ємний n -вимірний вектор $v = (v_1, v_2, \dots, v_n)$ такі, що виконуються умови

$$c + Dx - A^T \lambda^0 + y = 0, \quad (4.7)$$

$$b - Ax - w = 0, \quad (4.8)$$

$$(y, x) = 0, \quad (4.9)$$

$$(w, \lambda) = 0. \quad (4.10)$$

Співвідношення (4.7), (4.8) утворюють систему з $n + m$ лінійних рівнянь із $2(n + m)$ невідомими (координатами всіх векторів x, λ, v, w), а (4.9), (4.10) — це умови доповнювальної нежорсткості, які означають, що між координатами векторів x і v, λ та w існує така залежність: якщо $x_j^0 > 0$, то $v_j = 0$, і навпаки; якщо $\lambda_i^0 > 0$, то $w_j = 0$, і навпаки.

Така система лінійних рівнянь має безліч розв'язків. Оскільки нас цікавлять лише невід'ємні базисні розв'язки, то їх можна шукати за допомогою звичайного симплекс-методу, якщо сформулювати ЗЛП відносно якоїсь псевдоцільової функції. Найпростіше це зробити за допомогою штучних змінних у вигляді векторів $z = (z_1, z_2, \dots, z_n)$ та $y = (y_1, y_2, \dots, y_m)$, які потрібно ввести в умови (4.7), (4.8) із тим самим знаком, що й вектори c та b .

Отримуємо таку ЗЛП: знайти

$$M(y_1 + y_2 + \dots + y_m) + M(z_1 + z_2 + \dots + z_n) \rightarrow \min$$

для обмежень

$$\begin{cases} Dx - A\lambda + v + z = -c, \\ Ax + w + y = b. \end{cases}$$

У ході розв'язання такої задачі штучні змінні виводять із базису, а змінні x , λ , v , w — вводять. Якщо якийсь розв'язок x^0 , λ^0 , v^0 , w^0 задовольняє умови (4.9), (4.10), то він оптимальний, а не то слід продовжувати обчислення.

Приклад 4.5. Розв'язати задачу квадратичного програмування

$$\max f(x_1, x_2) = \max (32x_1 + 120x_2 - 4x_1^2 - 15x_2^2)$$

для обмежень

$$\begin{cases} 2x_1 + 5x_2 \leq 20, \\ 2x_1 - x_2 = 8, \\ x_1, x_2 \geq 0. \end{cases}$$

Оскільки матриця Гессе (других похідних) цільової функції $f(x_1, x_2)$

$$H = \begin{pmatrix} -8 & 0 \\ 0 & -30 \end{pmatrix}$$

від'ємно визначена (на головній діагоналі має визначники різних знаків), то цільова функція увігнута.

Складемо для цієї задачі функцію Лагранжа

$$\begin{aligned} L(x_1, x_2, \lambda_1, \lambda_2) &= (32x_1 + 120x_2 - 4x_1^2 - 15x_2^2) + \\ &+ \lambda_1(20 - 2x_1 - 5x_2) + \lambda_2(8 - 2x_1 + x_2). \end{aligned}$$

Умови теореми Куна — Таккера для цієї задачі мають вигляд

$$\left\{ \begin{array}{l} \frac{\partial L}{\partial x_1} = 32 - 8x_1 - 2\lambda_1 - 2\lambda_2 \leq 0, \\ \frac{\partial L}{\partial x_2} = 120 - 30x_2 - 5\lambda_1 + \lambda_2 \leq 0, \\ \frac{\partial L}{\partial \lambda_1} = 20 - 2x_1 - 5x_2 \geq 0, \\ \frac{\partial L}{\partial \lambda_2} = 8 - 2x_1 + x_2 = 0, \\ \frac{\partial L}{\partial x_1} x_1 = 0, \\ \frac{\partial L}{\partial x_2} x_2 = 0, \\ \frac{\partial L}{\partial \lambda_1} \lambda_1 = 0. \end{array} \right.$$

Оскільки друге обмеження задачі являє собою рівняння, то для нього не вводять умову доповнювальної нежорсткості, тобто змінна λ_2 не має обмеження на знак. Але для того, щоб не вийти із загальних умов теореми Куна — Таккера, вводять заміну $\lambda_2 = \lambda_3 - \lambda_4$, де $\lambda_3, \lambda_4 \geq 0$. За допомогою додаткових змінних v_1, v_2, w_1 систему обмежень перетворюємо до стандартного вигляду. Маємо систему рівнянь

$$\left\{ \begin{array}{lll} 32 - 8x_1 - & 2\lambda_1 - 2\lambda_2 + v_1 & = 0, \\ 120 - & 30x_2 - 5\lambda_1 + \lambda_2 + v_2 & = 0, \\ 20 - 2x_1 - 5x_2 - & & w_1 = 0, \\ & 8 - 2x_1 + x_2 & = 0 \end{array} \right.$$

й умови доповнювальної нежорсткості

$$x_1 v_1 = 0, \quad x_2 v_2 = 0, \quad \lambda_1 w_1 = 0.$$

Для пошуку її оптимального розв'язку сформулюємо ЗЛП: знайти

$$M(y_1, y_2, y_3) \rightarrow \min$$

для обмежень

$$\begin{cases} 8x_1 + 2\lambda_1 + 2\lambda_3 - 2\lambda_4 - v_1 + y_1 & = 32, \\ 30x_2 + 5\lambda_1 - \lambda_3 + \lambda_4 - v_2 + y_2 & = 120, \\ 2x_1 + 5x_2 + w_1 & = 20, \\ 2x_1 - x_2 + y_3 & = 8. \end{cases}$$

Розв'язавши цю ЗЛП симплекс-методом, отримаємо оптимальний план $x_1^0 = 5$, $x_2^0 = 2$, $\lambda_1^0 = 28/3$, $\lambda_2^0 = \lambda_3^0 - \lambda_4^0 = -40/3$, $\max f(x_1, x_2) = 240$.

Контрольні запитання та завдання

1. Які задачі називають задачами нелінійного програмування?
2. Формалізація яких економічних задач дає задачі нелінійного програмування?
3. Які задачі вирізняють серед ЗНП?
4. Дайте геометричну інтерпретацію ЗНП.
5. Коли застосовують метод множників Лагранжа?
6. Яка послідовність розв'язання задачі методом множників Лагранжа?
7. Означте поняття опуклої множини.
8. Як означають опуклу функцію?
9. Які задачі називають задачами опуклого (увігнутого) програмування?
10. Яку властивість мають ЗОП?
11. Які чисельні (ітераційні) методи застосовують для розв'язання задач безумовної оптимізації?
12. Які чисельні (ітераційні) методи застосовують для розв'язання задач умовної оптимізації?
13. Як застосовують теорію двоїстості до розв'язання задач умовної оптимізації?
14. Які задачі називають задачами квадратичного програмування?
15. Які методи застосовують для розв'язання задач квадратичного програмування?

5. ДИНАМІЧНЕ ПРОГРАМУВАННЯ

В економіці часто виникають ситуації, коли потрібно приймати рішення не одноразово, а крок за кроком. Одна з таких задач — розподіл капітальних вкладень, коли на певних етапах (наприклад, щорічно) потрібно розв'язувати задачу оптимального розподілу ресурсів. Зрозуміло, що рішення, прийняті на попередніх етапах, впливають на ті, що прийняті на наступних. Тому такі задачі не можна розглядати як окремі оптимізаційні задачі, а слід шукати послідовність взаємопов'язаних оптимальних рішень, що складуть загальне оптимальне рішення багатоетапної задачі. Для розв'язання таких задач потрібно застосовувати методи динамічного програмування.

Розв'язання задач динамічного програмування базується на *принципі оптимальності*, сформульованому математиком Р. Беллманом: якщо якась послідовність рішень оптимальна, то окремі наступні рішення всередині неї оптимальні відносно попередніх рішень або, інакше кажучи, оптимальна стратегія розв'язання багатоетапної задачі складається з пошуку оптимальних розв'язків її окремих етапів. Згідно з цим принципом задачу розв'язують, починаючи з останнього етапу. Потім переходять до передостаннього, відкидаючи всі можливі варіанти розв'язків, що суперечать уже відібраному рішення, і так далі, аж поки не буде знайдено оптимальне рішення початкового етапу.

Появу динамічного програмування зумовили насамперед динамічні задачі керування запасами. Це один із найчисленніших класів економічних задач. Правильне та своєчасне визначення оптимальної стратегії керування ресурсами, а також нормативного рівня їх використання дає змогу вивільнити значні оборотні кошти, заморожені у вигляді запасів, значно підвищити ефективність використання ресурсів і в остаточному підсумку підвищити ефективність усього виробництва загалом.

Розглянемо одну з типових задач динамічного програмування — розподіл капітальних вкладень.

Приклад 5.1. Нехай x — обсяг капітальних вкладень, які потрібно розподілити між двома підприємствами з різною прибутковістю. Обсяг капітальних засобів y , вкладених у перше підприємство, за рік дає прибуток $g(y) = 0,8y$. Частка капітальних засобів $x - y$, вкладених у друге підприємство, за рік дає прибуток $h(x - y) = 0,5(x - y)$. На кінець року засоби, вкладені в перше підприємство, становлять $a(y) = 0,3y$, а в друге — $b(x - y) = 0,6(x - y)$. Після закінчення кожного року засоби, що залишилися, знову розподіляють між підприємствами. Потрібно визначити такий розподіл, щоб загальний прибуток за три роки був максимальним.

Визначимо y_2 — обсяг засобів, вкладених у перше підприємство впродовж третього року, а потім y_1, y_0 — вкладення відповідно другого та першого років.

Якщо x_2 — кількість ресурсів, отриманих на початок третього року, то прибуток від обох підприємств — це сума $g(y_2) + h(x_2 - y_2)$. Потрібно визначити умови, за яких ця сума буде максимальною, тобто розв'язати задачу

$$\begin{aligned} f_1(x_2) &= \max_{0 \leq y_2 \leq x_2} \{g(y_2) + h(x_2 - y_2)\} = \max_{0 \leq y_2 \leq x_2} \{0,8y_2 + 0,5(x_2 - y_2)\} = \\ &= \max_{0 \leq y_2 \leq x_2} \{0,5x_2 + 0,3y_2\}. \end{aligned}$$

Очевидно, що максимум можливий за умови $x_2 = y_2$. Це означає, що всі ресурси на останньому етапі слід спрямувати в перше підприємство. При цьому буде отримано прибуток $f_1(x_2) = 0,8x_2$.

Тепер знайдемо y_1 . Розглянемо двокроковий процес — останній і передостанній етапи. У цей проміжок часу потрібно якнайкраще використати ресурси x_1 , не зважаючи на розподіл ресурсів на попередньому (першому) етапі. Максимальний загальний прибуток на останніх двох етапах становить

$$f_2(x_1) = \max_{0 \leq y_1 \leq x_1} \{g(y_1) + h(x_1 - y_1) + f_1[a(y_1) + b(x_1 - y_1)]\},$$

де $g(y_1) + h(x_1 - y_1)$ — прибуток на передостанньому етапі, $f_1[a(y_1) + b(x_1 - y_1)]$ — максимальний прибуток на останньому етапі, коли

загальна кількість ресурсів x_2 складається з решти залишків обох підприємств $x_2 = a(y_1) + b(x_1 - y_1)$.

Потрібно вибрати таке y_1 із проміжку $[0, x_1]$, щоб значення $f_2(x_1)$ було максимальним. Оскільки $f_1(x_2) = 0,8x_2$, то

$$\begin{aligned} f_2(x_1) &= \max_{0 \leq y_1 \leq x_1} \{0,8y_1 + 0,5(x_1 - y_1) + \\ &+ f_1[0,3y_1 + 0,6(x_1 - y_1)]\} = \max_{0 \leq y_1 \leq x_1} \{0,8y_1 + 0,5(x_1 - y_1) + \\ &+ 0,8 [0,6x_1 - 0,3y_1]\} = \max_{0 \leq y_1 \leq x_1} \{0,96x_1 + 0,06y_1\}, \end{aligned}$$

звідки $f_2(x_1) = 1,04x_1$ для $y_1 = x_1$. Це означає, що всі капітальні вкладення на передостанньому (другому) етапі потрібно спрямувати в перше підприємство.

Аналогічно, для першого етапу

$$\begin{aligned} f_3(x) &= \max_{0 \leq y \leq x} \{g(y) + h(x - y) + f_2[a(y) + b(x - y)]\} = \\ &= \max_{0 \leq y \leq x} \{0,8y_1 + 0,5(x - y) + 1,04 [0,3y + 0,6(x - y)]\} = \\ &= \max_{0 \leq y \leq x} \{1,124x - 0,012y\}. \end{aligned}$$

Максимум досягається в разі $y = 0$. Це означає, що на першому етапі всі ресурси потрібно спрямувати в друге підприємство. Отже, знайдено оптимальну стратегію: $y = 0$; $y_1 = x_1$; $y_2 = x_2$. Якщо дотримуватися її, то найбільший загальний прибуток на всіх етапах становитиме $f_3(x) = 1,124x$.

Дійсно, якщо на першому етапі всі ресурси спрямувати в друге підприємство, то буде отримано прибуток $h(x) = 0,5x$. На кінець року залишиться $b(x) = 0,6x$ капітальних вкладень, які в наступному (другому) році будуть спрямовані в перше підприємство, і це дасть прибуток $g(0,6x) = 0,8 \cdot 0,6x = 0,48x$. Залишок ресурсів при цьому становитиме $a(0,6x) = 0,3 \cdot 0,6x = 0,18x$. Прибуток на третьому (останньому) етапі дорівнюватиме $g(0,18x) = 0,8 \cdot 0,18x = 0,144x$. Загальний прибуток на всіх етапах становитиме $0,5x + 0,48x + 0,144x = 1,124x$.

Усі інші способи розподілу капітальних вкладень між підприємствами можуть лише зменшити загальний прибуток.

Інший приклад динамічної задачі — задача про використання робочої сили. Виконавцеві робіт потрібно визначити оптимальну чисельність працівників у кожний з n місяців. Виробничі завдання для кожного місяця відомі. Припустімо, що для виконання певного виду робіт j -го місяця потрібно m_j працівників. Якби можна було наймати та звільняти працівників без додаткових витрат, то можна було б кожного місяця мати їх рівно m_j . Однак реально для зміни чисельності працівників потрібні певні витрати, у поточному (x_j) та попередньому (x_{j-1}) місяці; їх задають функцією $f_j(x_j - x_{j-1})$. Знак її аргументу визначає тип витрат: якщо $x_j \geq x_{j-1}$, то $f_j(x_j - x_{j-1})$ — це витрати, пов'язані з найманням працівників, а в разі $x_j < x_{j-1}$ — зі звільненням.

Якщо з певних причин не виконано норми щодо чисельності працівників у поточному місяці, то підприємство має збитки $g_j(x_j - m_j)$, де m_j — “ідеальна” чисельність, навіть якщо виконання робіт забезпечене наднормативною працею. І лише тоді, коли $x_j = m_j$, витрати нульові, тобто $g_j(0) = 0$, $j = 1, 2, \dots, n$.

Щоб визначити оптимальну чисельність персоналу, потрібно мінімізувати цільову функцію

$$z = \sum_{j=1}^n [f_j(x_j - x_{j-1}) + g_j(x_j - m_j)]$$

за умови, що $x_j \geq 0$, $x_0 = m_0$. Розв'язання, як і для попередньої задачі, краще починати з останнього етапу (місяця). Позначимо витрати цього місяця як

$$\Lambda_n(x) = \min_{x_n} [f_n(x_n - x) + g_n(x_n - m_n)],$$

де $x = x_{n-1}$. Тоді мінімальні витрати k -го місяця можна обчислити за рекурентним співвідношенням

$$\Lambda_k(x) = \min_{x_k} [f_k(x_k - x) + g_k(x_k - m_k) + \Lambda_{k+1}(x_k)],$$

де $x = x_{k-1}$ — кількість працівників у $(k - 1)$ -му місяці. Для $k = 1$ визначають загальні витрати.

На останньому кроці визначають оптимальну чисельність працівників у першому місяці x_1 за умови, що на початок місяця їх було m_0 , а потім — послідовно кількість працівників x_2 через відоме x_1 , x_3 через відоме x_2 і т. д. до останнього n -го місяця. Такий підхід потрібен тому, що не відома кількість працівників у $(n + 1)$ -му місяці.

Якщо крім m_0 задано m_{n+1} , то потрібно шукати такі цілі числа x_j , що мінімізують цільову функцію

$$z = \sum_{j=1}^n [f_j(x_j - x_{j-1}) + g_j(x_j - m_j) + f_{n+1}(m_{n+1} - x_n)].$$

Тут

$$z = \min_{x_n} \left\{ f_{n+1}(m_{n+1} - x_n) + \min_{x_1, x_2, \dots, x_n} \sum_{j=1}^n [f_j(x_j - x_{j-1}) + g_j(x_j - m_j)] \right\}.$$

Оскільки в цій задачі задано кінцевий стан, тобто значення m_{n+1} , то її розв'язують у пряму напрямку. Основне рекурентне співвідношення такої задачі —

$$\Lambda_k(x) = \min_{x_k} [f_{k+1}(x_k - x) + g_{k+1}(x - m_k) + \Lambda_{k-1}(x_k)],$$

де функція $\Lambda_k(x)$ задає мінімальні витрати протягом перших $k + 1$ місяців за умови, що кількість працівників у $(k + 1)$ -й місяць дорівнює x .

Зауваження. Якщо задано початкові умови, то задачу динамічного програмування розв'язують у зворотному напрямку, а якщо кінцеві, то в пряму. Нарешті, якщо задано й початкові, і кінцеві умови, то її можна розв'язувати в будь-якому напрямку. Результати розв'язання за обома схемами однакові.

Наведемо ще декілька прикладів задач динамічного програмування.

У разі планування роботи системи постачання на n періодів зі змінним попитом виникають так звані динамічні задачі *керування запасами*. Витрати на виконання замовлення на певний вид ресурсів у кожному періоді залежать від загального обсягу замовлення цього ресурсу та його залишків від попередніх періодів. Крім того, витрати на збереження надлишків ресурсів залежать від різниці між наявною

кількістю ресурсу й обсягом попиту на нього в цьому періоді. Задача полягає в тому, щоб знайти такі обсяги замовлень відповідного ресурсу, щоб загальні витрати на його зберігання були мінімальними.

Такі задачі не мають аналітичного розв'язку, але їх можна розв'язувати методом динамічного програмування.

Багато задач оптимального планування та керування запасами можна подати у вигляді графа. Кожному стану системи відповідає якась вершина графа, а перехід від одного стану системи до іншого інтерпретують напрямленими дугами, кожній з яких поставлено у відповідність витрати, потрібні для виконання такого переходу. Тоді задача оптимального планування полягає в пошуку найкоротшого шляху в мережі.

У загальному випадку динамічне програмування — це обчислювальний метод для розв'язання задач певної структури, а саме тих, у яких цільова функція адитивна чи мультиплікативна, тобто складається із суми чи добутку функцій, кожна з котрих залежить лише від одного аргументу.

Нагадаємо, що функція багатьох змінних $f(x_1, x_2, \dots, x_n)$ називається *адитивною*, якщо $f(x_1, x_2, \dots, x_n) = \sum_{i=1}^n f_i(x_i)$, і *мультиплікативною*,

якщо $f(x_1, x_2, \dots, x_n) = \prod_{i=1}^n f_i(x_i)$. Оскільки логарифм від мультипліка-

тивної функції — адитивна функція, то зазвичай розглядають адитивні функції.

Формально задача динамічного програмування має такий вигляд: знайти

$$\max_{x_1, x_2, \dots, x_n} f(x_1, x_2, \dots, x_n) = \sum_{j=1}^n f_j(x_j)$$

за умов

$$\sum a_j x_j \leq b, \quad a_j > 0, \quad x_j \geq 0, \quad j = 1, 2, \dots, n.$$

Загалом задачу динамічного програмування можна інтерпретувати як проблему оптимального вкладення якихось j ресурсів, вимірних у одиницях одиницях (наприклад, грошах) за допомогою вагових

коефіцієнтів a_j , у різноманітні активи (інвестиційні проекти, підприємства та ін.), кожен з яких характеризується функцією прибутку f_j , тобто йдеться про такий розподіл обмеженого ресурсу b , який максимізує сумарний прибуток.

Динамічне програмування — це метод спрямованого перебору варіантів. Процес оптимізації тут розглядають як багатокроковий процес прийняття взаємопов'язаних рішень.

На кожному k -му кроці цього процесу обчислюють значення тільки однієї змінної x_k для всіх можливих значень внутрішніх параметрів задачі, що не змінюються від кроку до кроку, з урахуванням результатів, отриманих на попередніх етапах. Наприклад, у задачі розподілу капітальних вкладень (приклад 5.1) внутрішні параметри задачі — це норми прибутковості підприємств (незмінні показники), а значення x_2, x_1, x обчислюють на кожному етапі з урахуванням отриманих результатів та за умови оптимальності поточного рішення.

Для застосування схеми динамічного програмування потрібно, щоб було виконано такі передумови:

1) процес пошуку оптимального рішення слід розглядати як багатокроковий процес прийняття рішень;

2) внутрішні параметри задачі мають бути незмінними від кроку до кроку, але їх має бути декілька, щоб забезпечити альтернативність рішень;

3) оптимальні рішення поточного кроку не впливають на рішення, отримані на попередніх кроках;

4) оптимальний розв'язок поточного кроку залежить лише від поточних умов задачі, але не залежить від розв'язків, що спричинили саме такі умови, тобто не залежить від передісторії процесу.

Саме останній пункт забезпечує реалізацію принципу оптимальності Беллмана, який полягає в тому, що на кожному кроці шукають не ізольоване оптимальне рішення, а вибирають його так, щоб забезпечити оптимальність рішень на наступних кроках. Формально цей принцип реалізують, будуючи на кожному кроці умовні оптимальні рішення, що забезпечують найбільшу сумарну ефективність, починаючи з поточного кроку, за умови, що відомий стан, який утворився до початку поточного кроку.

Контрольні запитання та завдання

1. Які задачі називають задачами динамічного програмування?
2. Який принцип покладено в основу методів розв'язання задач динамічного програмування?
3. Який вигляд має цільова функція в задачі динамічного програмування?
4. Яким вимогам задовольняють задачі, розв'язувані методами динамічного програмування?
5. Яка головна ідея методу розв'язання задач динамічного програмування?
6. За якими ознаками визначають напрямок розв'язання задач динамічного програмування?
7. Наведіть приклади задач динамічного програмування.

6. ЗАДАЧІ ТА МЕТОДИ СТОХАСТИЧНОГО ПРОГРАМУВАННЯ

Детерміновані моделі математичного програмування часто виявляються неадекватними реальним процесам, що відбуваються в економіці. Це пояснюється перш за все неточністю та випадковістю показників унаслідок впливу зовнішнього середовища. Зрозуміло, що покладені в основу моделі параметри, які неконтрольовано змінюються з плином часу, значно впливають на розв'язки оптимізаційних задач. Нехтування цього призводить до великих відхилень прогнозованих результатів від дійсних. Рішення, прийняті на основі оптимістичних прогнозів факторів, що зумовлюють виробництво та споживання, часто виявляється недопустимим, бо немає резервів для корекції розбіжностей між планом і реальними показниками будь-якої ланки загального ланцюга виробництва. Інколи вдається планувати за середніми показниками, однак, якщо розкид значень показників досить великий, то реальні їх значення можуть суттєво відрізнятись від середніх. У цьому разі запропонований план також може виявитися непридатним.

Для розв'язання задач планування в умовах ризику та невизначеності розроблено спеціальні моделі та методи — стохастичне програмування.

Загальна задача математичного програмування полягає в мінімізації (максимізації) цільової функції $f(x)$ на якійсь допустимій множині значень X , заданій системою обмежень $g_i(x) \leq 0, i = 1, 2, \dots, m$. Задачі стохастичного програмування виникають у тому разі, коли функції $f(x)$ і $g_i(x)$ залежать від випадкових параметрів ω . При цьому припускають, що ω — елемент простору стану природи (зовнішнього середовища) Ω . Тоді *загальну задачу стохастичного програмування* формують так: знайти

$$f(\mathbf{x}, \omega) \rightarrow \min$$

для обмежень

$$g_i(\mathbf{x}, \omega) \leq 0, \quad i = 1, 2, \dots, m.$$

Постановки задач стохастичного програмування суттєво залежать від того, чи можливо спостерігати стан зовнішнього середовища до вибору остаточного рішення. При цьому розрізняють задачі оперативного та перспективного стохастичного програмування.

У задачах *оперативного* стохастичного програмування рішення приймають після якогось експерименту (спостереження) над станом зовнішнього середовища; воно залежить від результатів цього експерименту та являє собою випадковий вектор $\mathbf{x} = \mathbf{x}(\omega)$. Такі задачі виникають, наприклад, в оперативному техніко-економічному плануванні, у медичній діагностиці та ін.

У задачах *перспективного* стохастичного програмування рішення приймають до проведення експерименту над станом зовнішнього середовища, тому воно детерміноване. Такі задачі виникають у перспективному техніко-економічному плануванні, у проектуванні, коли параметри системи потрібно вибирати як цілком конкретні детерміновані величини в розрахунку на певний діапазон відхилень.

Існують два основні *підходи* до розв'язання задач стохастичного програмування:

1) *непрямі* методи, які полягають у знаходженні детермінованих цільової функції та функцій-обмежень і розв'язанні відповідної детермінованої ЗНП; такі підходи застосовують лише для обмеженого кола задач;

2) *прямі* методи стохастичного програмування, що ґрунтуються на інформації про значення випадкових функцій $f(\mathbf{x}, \omega)$ та $g_i(\mathbf{x}, \omega)$ внаслідок реалізації експериментів (спостережень).

Перший підхід застосовують тоді, коли в стабільній економічній ситуації розв'язки оптимізаційних задач навіть за різних зовнішніх впливів мало різняться між собою. Тоді оптимальні рішення приймають на основі відомих стохастичних характеристик розподілу випадкових параметрів умов задачі до спостереження за реалізаціями поточних значень цих параметрів. Прийняте таким способом рішення — найкраще середньостатистичне рішення, а відповідна задача називається *одноетапною* задачею стохастичного програмування.

Однак у більшості випадків такий підхід може виявитися неприйнятним через значні коливання зовнішніх умов.

Частіше застосовують другий підхід, коли враховують випадковий характер цільової функції та функцій обмежень. У цьому разі формально постановки задач стохастичного програмування мають певні відмінності; їх розрізняють за характером розв'язків, вибором показника якості рішення та способом розмежування обмежень задачі.

У задачах стохастичного програмування з імовірнісними обмеженнями цільову функцію та функції-обмеження подано у вигляді таких функціоналів, як математичне сподівання, дисперсія чи ймовірність перевищення певного порогу. Якщо цільову функцію подано через математичне сподівання, то такі моделі називаються *M-моделями*, якщо через дисперсії — то *V-моделями*. Якщо потрібно максимізувати ймовірність перевищення певного порогу, то такі моделі називають *P-моделями*. До цієї самої групи відносять задачі, у яких потрібно мінімізувати поріг, котрий не може перевищити цільова функція із заданою ймовірністю.

Двохетапні задачі стохастичного програмування. У ході дослідження багатьох задач планування й управління в разі неповної інформації можливо та доцільно розбити процес прийняття рішень на два етапи: на першому вибрати так званий попередній план (план-прогноз), що дає змогу провести потрібні підготовчі роботи, на другому — компенсувати розбіжності, виявлені після спостережень реальних значень випадкових параметрів умов задачі, тобто виробити оптимальний план-компенсацію.

Природно, що план-прогноз і план-компенсація мають бути узгоджені, щоб забезпечити мінімум середнього значення сумарних витрат, які виникають на обох етапах розв'язання задачі, тобто в задачі, що має розв'язок, вибір попереднього плану має гарантувати існування плану-компенсації.

Приклад 6.1. Нехай підприємство виробляє n видів продукції, використовуючи m видів ресурсів. Замовлення на постачання ресурсів подають раніше, ніж точно відомі замовлення споживачів на кінцеву продукцію підприємства, а ϵ лише ймовірний прогноз попиту споживачів. Після уточнення попиту складають остаточний план виробництва з урахуванням таких можливостей:

- 1) можливої додаткової купівлі ресурсів за підвищеними цінами;
- 2) можливої відмови від частини замовлених ресурсів, пов'язану з певним штрафом;

3) неповного виконання запланованих продажів власної продукції та додаткових витрат при цьому;

4) неповного забезпечення попиту споживачів на вироблену продукцію та збитків у вигляді штрафу.

Нехай відомі норма використання i -го ресурсу на випуск j -го виробу a_{ij} ; обсяг замовлень x_i на ресурс i -го типу та його ціна c_i ; обсяг додатково закупленого ресурсу y_i^1 та його нова (вища) ціна d_i ; обсяг y_i^2 ресурсу i -го типу, від якого підприємство відмовляється, та штраф f_i , який воно сплатить за відмову від одиниці цього ресурсу; планований обсяг випуску власної продукції w_j ; випадковий вектор $\{\xi_j\}_{j=1}^n$ попиту на продукцію підприємства; дохід b_j від реалізації одиниці продукції j -го типу, а також штраф g_j за недопоставку одиниці продукції відповідного типу та витрати, які понесе підприємство в разі нереалізації власної продукції h_j .

Таку задачу розв'язують у два етапи. Спершу виробляють план-прогноз, тобто визначають обсяги замовлень x_i на потрібні у виробництві ресурси, які б забезпечили максимальний прибуток підприємства від реалізації власної продукції в запланованих обсягах w_j . На наступному етапі після надходження замовлення на продукцію підприємства складають план-компенсацію, тобто визначають, як слід скоригувати замовлення на ресурси, щоб зменшити збитки підприємства, пов'язані з відмовою від частини власних замовлень на ресурси та закупівлею додаткових ресурсів за підвищеними цінами, а також із невиконанням замовлень на власну продукцію. Інакше кажучи, після визначення вектора $\{\xi_j\}_{j=1}^n$ потрібно знайти такі y_i^1 , y_i^2 , щоб витрати підприємства були якнайменшими.

Формально розглядають таку задачу: знайти $f(\mathbf{c}, \mathbf{x}) \rightarrow \min$ за умов

$$A\mathbf{x} = \mathbf{b}, \quad (6.1)$$

$$A^{(1)}\mathbf{x} = \mathbf{b}^{(1)}, \quad (6.2)$$

$$\mathbf{x} \geq \mathbf{0}. \quad (6.3)$$

Тут \mathbf{c} , \mathbf{x} — вектори вартості й обсягу замовлених ресурсів i -го типу (обидва з розмірністю m), A й $A^{(1)}$ — матриці з розмірністю $n \times m$, які задають норми використання ресурсів на виробництво одиниці про-

дукції, \mathbf{b} — прибуток від реалізації продукції підприємства (вектор з розмірністю n).

Матриця $A = A(\omega)$ та вектори $\mathbf{b} = \mathbf{b}(\omega)$, $\mathbf{c} = \mathbf{c}(\omega)$ залежать від стану зовнішнього середовища ω ; це випадкові величини. Рішення \mathbf{x} потрібно приймати до спостереження за станом зовнішнього середовища, тобто до визначення параметрів ω . Для цього використовують детерміновані матрицю $A^{(1)}$ і вектор $\mathbf{b}^{(1)}$.

Нехай $\tilde{\mathbf{x}}$ — розв'язок, що задовольняє умови (6.2) та (6.3), отриманий для фіксованого значення $\tilde{\omega}$ — якоїсь реалізації випадкової величини ω (розв'язок першого етапу). На другому етапі потрібно оцінити відхил $\mathbf{b}(\tilde{\omega}) - A(\tilde{\omega})\tilde{\mathbf{x}}$ за умов (6.1) та обчислити вектор $\mathbf{y} \geq \mathbf{0}$ ($\mathbf{y} = y^1, y^2$), що його компенсує відповідно до співвідношення $D\mathbf{y} = \mathbf{b}(\tilde{\omega}) - A(\tilde{\omega})\tilde{\mathbf{x}}$, де $D_{n \times m_1}$ — матриця компенсації (узагалі кажучи, випадкова).

Матрицю A інтерпретують як матрицю основних технологічних способів виробництва (норм використання ресурсів), а D — як матрицю аварійних технологічних засобів, від яких залежать можливі способи компенсації виявлених нев'язок.

За порушення умов задачі накладають штраф, що залежить від вектора \mathbf{y} ; кожну його компоненту оцінюють величиною $q_k \geq 0$, $k = 1, 2, \dots, m_1$.

Вектор \mathbf{y} вибирають так, щоб мінімізувати можливі втрати підприємства, тобто розв'язують таку задачу:

$$(\mathbf{q}, \mathbf{y}) \rightarrow \min, \quad (6.4)$$

$$D\mathbf{y} = \mathbf{b} - A\mathbf{x}, \quad (6.5)$$

$$\mathbf{y} \geq 0. \quad (6.6)$$

Задачу (6.4)–(6.6) розв'язують на другому етапі.

Обидва етапи можна звести до одного запису й отримати таку задачу стохастичного програмування: знайти

$$\min_x M_\omega \left[(\mathbf{c}(\omega), \mathbf{x}) + \min (\mathbf{q}(\omega), \mathbf{y}) \right]$$

за умов

$$A^{(1)}x = b^{(1)},$$

$$D(\omega)y = b(\omega) - A(\omega)x,$$

$$x \geq 0.$$

Тут M_ω — математичне сподівання цільової функції відносно параметрів зовнішнього середовища ω .

Розв'язок зазначеної задачі складається з двох векторів: детермінованого m -вектора x і випадкового m_1 -вектора $y(\omega)$, які визначають відповідно попередній план і план-компенсацію.

Для розв'язання такої задачі спершу обчислюють вектор x , а після реалізації ω — випадкових параметрів зовнішнього середовища — шукають y як розв'язок ЗЛП (6.4)–(6.6).

Оскільки витрати на корекцію плану мають бути якомога меншими, то для перспективного планування доцільно застосовувати малочутливі до зміни параметрів умови задачі.

Для існування розв'язку задачі стохастичного програмування потрібно, щоб розв'язки обох етапів були узгоджені між собою. Доведено, що для розв'язності задачі другого етапу (6.4)–(6.6) для будь-яких реалізацій A та будь-якого плану x необхідно й достатньо, щоб система нерівностей $Dz \leq q$ мала хоча б один розв'язок.

Зауваження. Для двохетапної задачі стохастичного програмування можна побудувати еквівалентну детерміновану задачу, але це буде ЗОП, яку розв'язують методами, придатними для цього типу задач.

Контрольні запитання та завдання

1. Коли виникають задачі стохастичного програмування?
2. На які класи поділяють задачі стохастичного програмування?
3. Коли задачі стохастичного програмування розглядають як одноетапні?
4. Які методи застосовують для розв'язання одноетапних задач стохастичного програмування?
5. Коли задачі стохастичного програмування розглядають як двохетапні?
6. Які методи застосовують для розв'язання двохетапних задач стохастичного програмування?

7. МАТРИЧНІ ІГРИ

Задачі попередніх розділів було розглянуто для ситуацій індивідуального вибору оптимальних рішень, тобто коли рішення приймає суб'єкт, який керується єдиною метою. Принципово інша ситуація виникає в тому разі, коли рішення приймають декілька суб'єктів, інтереси яких не збігаються чи взагалі суперечливі. При цьому виникають задачі з багатьма цільовими функціями (критеріями). Крім того, унаслідок дій двох або більше “розумних” супротивників такі задачі мають невизначеність, бо кожен з учасників процесу прийняття рішень може оптимізувати свою поведінку за рахунок інших або ж відійти від бажаних рішень через спротив інших учасників.

Одне з основних питань у задачах колективного прийняття рішень — оптимальність: потрібно визначити, які рішення оптимальні, коли існує декілька критеріїв, які відображають різні інтереси.

Розділ математичного програмування, що вивчає такі задачі, називається *теорією ігор*.

Термін “гра” застосовують для позначення сукупності правил і угод, якими керуються суб'єкти, приймаючи колективне рішення. Кожен такий суб'єкт (гравець) має індивідуальну систему цільових установок і стратегій — можливих варіантів індивідуальних рішень.

Розглянемо найпростіший варіант матричної гри з двома гравцями (конкурентами) J_1, J_2 й такими правилами. Перший гравець вибирає один з m можливих варіантів. Нехай це буде варіант i ($i = 1, 2, \dots, m$). Другий гравець може вибрати один з n варіантів, наприклад j ($j = 1, 2, \dots, n$). Варіанти i та j кожен із гравців вибирає самостійно. Результати всіх можливих ходів обох гравців заносять у так звану *платіжну матрицю*, або *матрицю вигравів* (для визначеності будемо вважати, що її складено для другого гравця) $C = \{c_{ij}\}$. Кожен елемент цієї матриці c_{ij} показує суму виграву другого гравця

за умови, що гравці J_1, J_2 вибрали відповідно варіанти i, j . Якщо $c_{ij} > 0$, то перший платить другому, а в разі $c_{ij} < 0$ — навпаки.

Ігри, у яких виграш одного учасника дорівнює програшу інших, називаються *іграми з нульовою сумою*. Якщо в таких іграх тільки два учасники, то ігри називаються *антагоністичними*.

Приклад 7.1 (угадкування монет). Кожен із гравців вибирає певний бік монети, потім гравці одночасно називають свій вибір. Якщо вибрано різні боки, то перший платить другому одну грошову одиницю; якщо однакові, то другий платить першому також одну грошову одиницю.

Оскільки кожен із гравців у цій грі має два варіанти вибору рішень, то платіжна матриця має вигляд

$$C = \begin{pmatrix} -1 & 1 \\ 1 & -1 \end{pmatrix}.$$

Нехай вектор $\mathbf{x} = (x_1, x_2, \dots, x_m)$ містить імовірності вибору першим гравцем кожного з m доступних йому варіантів, причому $\sum_{i=1}^m x_i = 1$, $0 \leq x_i \leq 1$, а вектор $\mathbf{y} = (y_1, y_2, \dots, y_m)$ — імовірності вибору кожного з n варіантів, доступних другому гравцеві, причому $\sum_{j=1}^n y_j = 1$, $0 \leq y_j \leq 1$.

Якщо гравець вибирає лише один із можливих варіантів рішень, тобто для якоїсь стратегії $\mathbf{x} = (x_1, \dots, x_i, \dots, x_m)$ виконується рівність $x_i = 1$, а всі інші $x_k = 0$ ($k = 1, 2, \dots, m, k \neq i$), то ця стратегія називається *i -ю чистою стратегією* гравця J_1 . Аналогічно означають *j -ту чисту стратегію* гравця J_2 . В усіх інших випадках стратегії називають *мішаними*.

Оптимальні чисті стратегії

Розглянемо матричну гру з позицій другого гравця. Нехай гравець J_2 вибирає варіант j . Це означає, що всі його можливі виграші записано в j -му стовпці платіжної матриці $C = \{c_{ij}\}$, тобто в стовпці $C_j = (c_{1j}, \dots, c_{mj})$. Щоб отримати гарантований виграш в умовах протидії іншого гравця, він має вибрати той варіант, де найменший еле-

мент у стовпці максимальний. Цей виграш називають *нижньою ціною гри* та позначають $\underline{v} = \max_{1 \leq j \leq n} \min_{1 \leq i \leq m} c_{ij}$.

Якщо замість C розглядати матрицю $-C = \{-c_{ij}\}$, то, очевидно, отримаємо матрицю виграшів першого гравця. Гравець J_1 також намагається гарантувати собі певний виграш і для цього вибирає такий варіант i (рядок платіжної матриці $-C$), у якому найменший елемент максимальний. Цей виграш становить $\max_{1 \leq i \leq m} \min_{1 \leq j \leq n} (-c_{ij}) = -\min_{1 \leq j \leq m} \max_{1 \leq i \leq n} c_{ij}$.

Тоді другий гравець отримає щонайбільше $\bar{v} = \min_{1 \leq i \leq m} \max_{1 \leq j \leq n} c_{ij}$. Величина \bar{v} називається *верхньою ціною гри*.

Отже, гравець J_2 може гарантувати собі щонайменше виграш \underline{v} , а гравець J_1 заважає йому отримати більше, ніж \bar{v} . Якщо існує таке компромісне рішення, що $v = \underline{v} = \bar{v}$, або

$$v = \max_i \min_j c_{ij} = \max_j \min_i c_{ij}, \quad (7.1)$$

то це означає, що гравець J_2 отримає щонайменше виграш v , але не більший, ніж v , через протидію свого конкурента. Варіанти рішень i^*, j^* , вибрані відповідно гравцями J_1, J_2 , за яких було досягнуто компромісу $v = (c_{i^* j^*})$ у співвідношенні (7.1), природно назвати *оптимальними рішеннями*. Тоді v називається *ціною гри*, а сама гра має розв'язок у чистих стратегіях.

Вектори x та y , що складаються з усіх нульових елементів, крім одного, який дорівнює одиниці ($x_{i^*} = 1, y_{j^*} = 1$), називаються *оптимальними чистими стратегіями* гравців J_1, J_2 .

Виявляється, що співвідношення (7.1) виконується не для кожної гри, заданої платіжною матрицею C , тобто не кожна матрична гра має розв'язок у чистих стратегіях. Наприклад, для гри “вгадування монет” із платіжною матрицею $C = \begin{pmatrix} -1 & 1 \\ 1 & -1 \end{pmatrix}$, очевидно, виконуються

співвідношення $\underline{v} = \max_j \min_i c_{ij} = -1, \bar{v} = \min_i \max_j c_{ij} = 1$, тобто $\underline{v} \neq \bar{v}$, і тому матрична гра не має розв'язку в чистих стратегіях.

Відповідь на запитання про розв'язність матричної гри в чистих стратегіях дає таке твердження.

Теорема 7.1. Матрична гра двох осіб із платіжною матрицею $C = \{c_{ij}\}$, $i = 1, 2, \dots, m, j = 1, 2, \dots, n$, має розв’язок у чистих стратегіях тоді й тільки тоді, коли матриця C має сідлову точку. Якщо (i^*, j^*) — сідлова точка матриці C , то ціна гри $v = c_{i^* j^*}$.

Зауваження. (i^*, j^*) називають сідловою точкою матриці C , якщо для всіх можливих значень i та j виконується співвідношення $c_{i^* j} \leq c_{i^* j^*} \leq c_{ij^*}$.

Якщо платіжна матриця гри має сідлову точку, то говорять, що й сама матрична гра має сідлову точку.

Приклад 7.2. Матриця вигравшів гравця J_2 має вигляд

$$C = \begin{array}{cc} & \max_j c_{ij} \\ & \begin{array}{c} 64 \\ 90 \\ 18 \\ 95 \end{array} \\ \begin{array}{c} 50 \\ 50 \\ 18 \\ 25 \end{array} & \begin{array}{ccc} 27 & 64 & \\ 5 & 90 & \\ 9 & 12 & \\ 95 & 20 & \end{array} \\ \min_i c_{ij} & \begin{array}{ccc} 18 & 5 & 12 \end{array} \end{array}$$

У додатковому нижньому рядку записано мінімальні значення стовпців матриці, а в додатковому правому стовпці — максимальні значення в рядках. Найбільше значення в додатковому рядку, що дорівнює 18, збігається з найменшим значенням у додатковому стовпці. Це означає, що в цьому прикладі $i^* = 3, j^* = 1$, тобто $(3; 1)$ — сідлова точка матриці C , а матрична гра має ціну $v = 18$. Якщо гравець J_1 відступить від своєї оптимальної стратегії $i^* = 3$, а гравець J_2 дотримуватиметься оптимальної стратегії $j^* = 1$, то перший програє більше ніж 18 одиниць. Аналогічно, якщо гравець J_2 відступить від своєї оптимальної стратегії, а J_1 дотримуватиметься оптимальної стратегії, то другий гравець виграє менше ніж 18 одиниць.

Оптимальні мішані стратегії

Якщо в платіжній матриці немає сідлової точки (наприклад, у грі “вгадування монет”), то жодному з гравців, очевидно, не варто дотримуватися тільки однієї стратегії, тому що в разі повторення гри противник адекватно реагуватиме на таку поведінку. Тоді природно внести елемент випадковості у виборі варіанта. Однак не слід робити

вибір хаотичним, бо це в середньому не дасть найліпшого результату. Доцільно віддавати більшу перевагу “ліпшим” варіантам (із можливих), а гірші застосовувати рідше. Тут потрібно перейти до мішаних стратегій (де задано ймовірності вибору кожного з доступних гравцям варіантів) й означити поняття оптимальної стратегії в цьому класі.

Нагадаємо, що мішаними стратегіями гравців називають упорядковані набори чисел (вектори), у яких кожен елемент інтерпретують як ймовірність вибору відповідних варіантів рішень. Для першого гравця це вектор $\mathbf{x} = (x_1, x_2, \dots, x_m)$ із такими елементами $0 \leq x_i \leq 1$, що

$$\sum_{i=1}^m x_i = 1, \text{ а для другого — вектор } \mathbf{y} = (y_1, y_2, \dots, y_n) \text{ із такими елементами}$$

$0 \leq y_j \leq 1$, що $\sum_{j=1}^n y_j = 1$. Тоді математичне сподівання платежу

гравця J_1 своєму суперникові — гравцю J_2 (середній виграш гравця J_2) становить

$$F(\mathbf{x}, \mathbf{y}) = \mathbf{x}C\mathbf{y} = \sum_{i=1}^m \sum_{j=1}^n x_i c_{ij} y_j. \quad (7.2)$$

Як і в разі чистих стратегій, гравець J_1 може забезпечити собі не більший виграш, ніж

$$\min_{\mathbf{x}} G(\mathbf{x}) = \min_{\mathbf{x}} \max_{\mathbf{y}} F(\mathbf{x}, \mathbf{y}), \quad (7.3)$$

а гравець J_1 — не менший, ніж

$$\max_{\mathbf{y}} H(\mathbf{y}) = \max_{\mathbf{y}} \min_{\mathbf{x}} F(\mathbf{x}, \mathbf{y}). \quad (7.4)$$

Співвідношення (7.3), (7.4) — це формальний запис задач пошуку мішаних стратегій, які забезпечать гарантований виграш кожному з гравців.

Якщо для якихось фіксованих мішаних стратегій \mathbf{x}^* й \mathbf{y}^* та для всіх інших мішаних стратегій \mathbf{x}, \mathbf{y} виконується співвідношення

$$F(\mathbf{x}^*, \mathbf{y}) \leq F(\mathbf{x}^*, \mathbf{y}^*) \leq F(\mathbf{x}, \mathbf{y}^*),$$

тобто (x^*, y^*) — сідлова точка функції $F(x, y)$, то доведено, що

$$F(x^*, y^*) = \min_x \max_y F(x, y) = \max_y \min_x F(x, y).$$

Отже, якщо функція $F(x, y)$, яка являє собою середній виграш другого гравця та задана співвідношенням (7.2), має сідлову точку, то можна ввести такі означення. Якщо існує сідлова точка (x^*, y^*) функції $F(x, y)$, то її компоненти x^* й y^* називаються відповідно *оптимальними стратегіями* гравців J_1 і J_2 , а значення самої функції $F(x^*, y^*)$ — *ціною гри*. При цьому також говорять, що матрична гра має розв'язок у мішаних стратегіях.

Доведено, що пошук сідлової точки (x^*, y^*) функції $F(x^*, y^*) = xCy$ пов'язаний із розв'язанням двоїстих ЗЛП; крім того, ця точка завжди існує, а це означає, що будь-яка матрична гра має розв'язок у мішаних стратегіях. Сформулюємо ці результати у вигляді теорем.

Теорема 7.2. Задачі (7.3), (7.4) пошуку гарантованих рішень гравців J_1 і J_2 еквівалентні таким ЗЛП:

$$\left. \begin{array}{l} x_{m+1} \rightarrow \min \\ \sum_{i=1}^m c_{ij} x_i \leq x_{m+1}, \quad j = 1, 2, \dots, n, \\ \sum_{i=1}^m x_i = 1, \quad x_i \geq 0, \quad i = 1, 2, \dots, m; \end{array} \right\} \quad (7.5)$$

$$\left. \begin{array}{l} y_{n+1} \rightarrow \max \\ \sum_{j=1}^n c_{ij} y_j \geq y_{n+1}, \quad i = 1, 2, \dots, m, \\ \sum_{j=1}^n y_j = 1, \quad y_j \geq 0, \quad j = 1, 2, \dots, n. \end{array} \right\} \quad (7.6)$$

Теорема 7.3 (основна теорема матричних ігор). Будь-яка матрична гра має розв'язок у мішаних стратегіях.

Приклад 7.3. Розглянемо гру “вгадування монет” із платіжною матрицею $C = \begin{pmatrix} -1 & 1 \\ 1 & -1 \end{pmatrix}$. Нехай $\mathbf{x} = (x_1, x_2)$, $\mathbf{y} = (y_1, y_2)$ — відповідно мішані стратегії гравців J_1 і J_2 . Середній виграш другого гравця становить

$$F(\mathbf{x}, \mathbf{y}) = \mathbf{x}C\mathbf{y} = y_1(-x_1 + x_2) + y_2(x_1 - x_2).$$

Задачі (7.3), (7.4) набирають вигляду

$$\max_{\mathbf{y}} [y_1(-x_1 + x_2) + y_2(x_1 - x_2)] \rightarrow \min_{\mathbf{x}};$$

$$\min_{\mathbf{x}} [y_1(-x_1 + x_2) + y_2(x_1 - x_2)] \rightarrow \max_{\mathbf{y}},$$

де $\mathbf{x} = (x_1, x_2)$, $x_1 \geq 0$, $x_2 \geq 0$, $x_1 + x_2 = 1$; $\mathbf{y} = (y_1, y_2)$, $y_1 \geq 0$, $y_2 \geq 0$, $y_1 + y_2 = 1$;
Задачі (7.6) і (7.7) можна записати у вигляді

$$\begin{array}{ll} x_2 \rightarrow \min, & y_3 \rightarrow \max, \\ \left\{ \begin{array}{l} -x_1 + x_2 \leq x_3, \\ x_1 - x_2 \leq x_3, \\ x_1 \geq 0, \quad x_2 \geq 0; \end{array} \right. & \left\{ \begin{array}{l} -y_1 + y_2 \geq y_3, \\ y_1 - y_2 \geq y_3, \\ y_1 \geq 0, \quad y_2 \geq 0 \end{array} \right. \end{array}$$

Розв’язавши ці двоїсті ЗЛП (наприклад, за допомогою симплекс-методу), отримаємо оптимальні змішані стратегії $\mathbf{x}^* = (1/2; 1/2)$; $\mathbf{y}^* = (1/2; 1/2)$ гравців J_1 і J_2 . При цьому ціна гри $F(\mathbf{x}^*, \mathbf{y}^*) = 0$.

Як було зазначено для чистих стратегій, якщо один із гравців дотримується своєї оптимальної стратегії, а другий відступає від своєї оптимальної стратегії, то останній гравець (той, що відступив) або виграє менше, або програє більше. Аналогічні дії гравців у разі мішаних стратегій дають результат, сформульований у теоремі 7.4.

Означення 7.1. Якусь i -ту стратегію гравця J_1 називають **активною**, якщо відповідна компонента оптимальної мішаної стратегії $\mathbf{x}^* = (x_1^*, \dots, x_m^*)$ строго більша нуля, тобто $x_i^* > 0$.

Аналогічно означають активні стратегії гравця J_2 .

Теорема 7.4. (про активні стратегії). Якщо гравець J_2 дотримується своєї оптимальної стратегії \mathbf{y}^* , то його середній виграш

$F(x, y^*) = xCy^*$ залишається незмінним і дорівнює ціні гри $v = F(x^*, y^*)$ незалежно від стратегії гравця J_1 , якщо тільки перший гравець (J_1) не виходить за межі своїх активних стратегій (користується будь-якою з них або змішує їх у будь-яких пропорціях).

Зауваження. Останню теорему можна переформулювати, помінявши гравців місцями.

Приклад 7.4. Нехай платіжна матриця гри $C = \begin{pmatrix} -2 & 1 \\ 1 & -1 \end{pmatrix}$. Знайти

оптимальний розв'язок задачі.

Насамперед потрібно дослідити матрицю C на наявність сідлової точки.

$$\begin{array}{l} \max_j c_{ij} \\ C = \begin{pmatrix} -2 & 1 \\ 1 & -1 \end{pmatrix} \\ \min_i c_{ij} \end{array} \begin{array}{l} 1 \\ 1 \\ -2 \quad -1 \end{array}$$

Очевидно, що платіжна матриця не має сідлової точки, тому задача не має розв'язку в чистих стратегіях. Будемо шукати його в класі мішаних стратегій. Нехай $y^* = (y_1^*, y_2^*)$ — оптимальна мішана стратегія гравця J_2 . Тоді вона має задовольняти системі рівнянь

$$\begin{cases} -2y_1^* + y_2^* = v, \\ y_1^* - y_2^* = v, \\ y_1^* + y_2^* = 1, \end{cases}$$

із якої визначаємо $y^* = (2/5; 3/5)$; $v = -1/5$.

Аналогічно знаходимо оптимальну змішану стратегію гравця J_1 : $x^* = (2/3; 3/5)$.

Графічні методи розв'язання матричних ігор

Іноді нерационально застосовувати стандартні методи ЛП, зокрема тоді, коли один із гравців має лише дві стратегії, тобто платіжна

матриця має розмірність $2 \times n$ або $m \times 2$. Такі задачі можна розв'язувати графічно.

Нехай (для визначеності) гравець J_1 може вибирати між двома стратегіями, які він використовує з імовірностями x_1 та $x_2 = 1 - x_1$. Тоді сподівані виграші, що відповідають чистим стратегіям гравця J_2 , можна задати рівняннями

$$z = c_{11}x_1 + c_{21}(1 - x_1), \quad z = c_{12}x_1 + c_{22}(1 - x_1), \dots,$$

$$z = c_{1n}x_1 + c_{2n}(1 - x_1),$$

або

$$z = (c_{11} - c_{21})x_1 + c_{21}, \quad z = (c_{12} - c_{22})x_1 + c_{22}, \dots, (c_{1n} - c_{2n})x_1 + c_{2n}.$$

Отже, сподівані виграші — лінійні функції від змінної $x_1 \in [0; 1]$, і їх можна відобразити на площині zOx_1 . Тоді значенням мінімального гарантованого виграшу першого гравця відповідає нижня ламана, складена з відрізків зазначених ліній. Згідно з принципом максиміну, оптимальному вибору першого гравця відповідає найвища точка цієї ламаної.

Приклад 7.5. Нехай платіжна матриця гри $C = \begin{pmatrix} 7 & 9 & 8 \\ 10 & 6 & 9 \end{pmatrix}$. Тоді

сподівані виграші першого гравця можна задати рівняннями

$$z = (7 - 10)x_1 + 10; \quad z = (9 - 6)x_1 + 6;$$

$$z = (8 - 9)x_1 + 9 \quad (\text{рис. 7.1}).$$

Найвищу точку нижньої ламаної утворено перетином першої та другої ліній; вона має координати $x_1 = 2/3$; $z = 8$. Тоді оптимальна мішана стратегія першого гравця — це $x^* = (2/3; 1/3)$, а ціна гри $v = z = 8$. Оптимальну мішану стратегію другого гравця знайдемо із системи рівнянь

$$\begin{cases} Cy = V, \\ \sum_{j=1}^3 y_j = 1, \end{cases}$$

де $V = (v, v)$, яка в координатній формі має вигляд

$$\begin{cases} 7y_1 + 9y_2 + 8y_3 = 8, \\ 10y_1 + 6y_2 + 9y_3 = 8, \\ y_1 + y_2 + y_3 = 1. \end{cases}$$

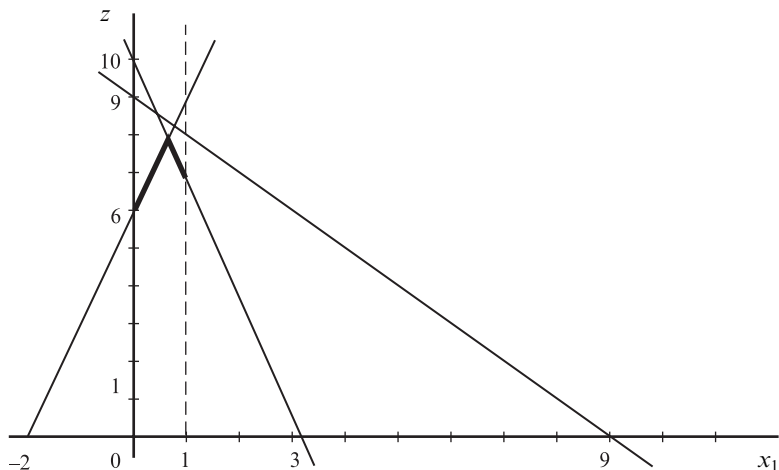


Рис. 7.1

Розв'язок цієї системи $y^* = (1/2; 1/2; 0)$ визначає оптимальну мішану стратегію другого гравця.

Аналогічно можна знайти графічний розв'язок задачі для випадку, коли другий гравець вибирає між двома стратегіями, тобто $y = (y_1, y_2)$, причому $y_2 = 1 - y_1$. Тоді сподівані виграші першого гравця можна задати прямими

$$z = c_{11}y_1 + c_{12}(1 - y_1), \quad z = c_{21}y_1 + c_{22}(1 - y_1), \dots,$$

$$z = c_{m1}y_1 + c_{m2}(1 - y_1),$$

або

$$z = (c_{11} - c_{12})y_1 + c_{12}, \quad z = (c_{21} - c_{22})y_1 + c_{22}, \dots,$$

$$z = (c_{m1} - c_{m2})y_1 + c_{m2}.$$

Ці лінії потрібно відобразити на площині zOy_1 . Тоді гарантованим виграшам другого гравця згідно з принципом мінімаксу відповідає верхня ламана, що складається з відрізків зазначених прямих.

Приклад 7.6. Нехай платіжна матриця гри $C = \begin{pmatrix} 6 & 5 \\ 2 & 7 \\ 1 & 8 \end{pmatrix}$. Тоді споді-

вані виграші першого гравця можна задати лінійними функціями

$$z = (6 - 5)y_1 + 5; \quad z = (2 - 7)y_1 + 7; \quad z = (1 - 8)y_1 + 8.$$

Найнижчу точку верхньої ламаної лінії утворено перетином першої та третьої прямих; вона має координати $y_1 = 3/8$; $z = 43/8$ (рис. 7.2).

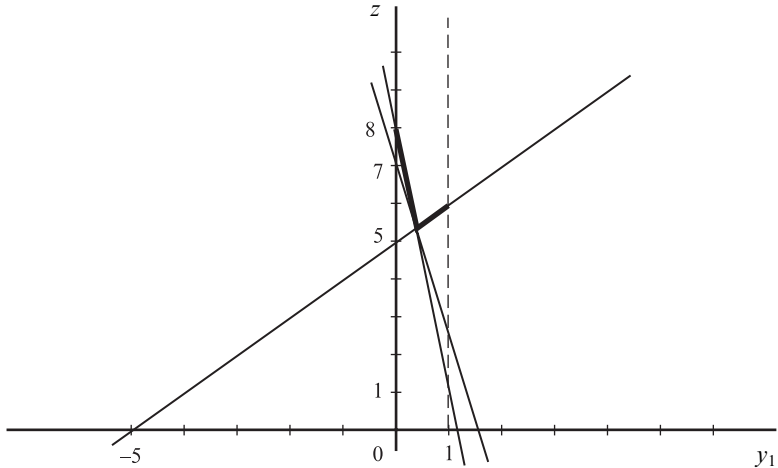


Рис. 7.2

Отже, оптимальна мішана стратегія другого гравця $y^* = (3/8; 5/8)$, а ціна гри становить $v = z = 43/8$. Оптимальну мішану стратегію першого гравця визначимо із системи рівнянь

$$\begin{cases} xC = V, \\ \sum_{i=1}^3 x_i = 1, \end{cases}$$

де $V = (v, v)$. Маємо

$$\begin{cases} 6x_1 + 2x_2 + x_3 = 43/8, \\ 5x_1 + 7x_2 + 8x_3 = 43/8, \\ x_1 + x_2 + x_3 = 1. \end{cases}$$

Розв'язок цієї системи $x^* = (7/8; 0; 1/8)$ — оптимальна мішана стратегія гравця J_1 . Ціна гри $v = 43/8$.

Хоча графічний метод застосовний лише до обмеженого кола задач, він добре відображає змістовний бік процесу пошуку оптимального розв'язку матричної гри.

Контрольні запитання та завдання

1. Які економічні ситуації розглядають як матричну гру?
2. Що називають матричною грою?
3. Що називають платіжною матрицею?
4. Як визначають стратегію гравця?
5. Яку стратегію називають чистою?
6. Яку стратегію називають мішаною?
7. Коли існує розв'язок матричної гри в чистих стратегіях?
8. Коли існує розв'язок матричної гри в мішаних стратегіях?
9. Як геометрично розв'язують матричну гру, якщо хоча б в одного гравця є тільки дві стратегії?
10. Як звести матричну гру до ЗЛП?

8. РОЗВ'ЯЗУВАННЯ ДЕЯКИХ ЗАДАЧ МАТЕМАТИЧНОГО ПРОГРАМУВАННЯ З ВИКОРИСТАННЯМ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Розв'язування задач із використанням програмного засобу GRAN1

Більшість задач математичного програмування можна розв'язувати графічно. Для графічного розв'язування задач на площині доречно використовувати програму GRAN1 (автори М. І. Жалдак, Ю. В. Горошко, НПУ ім. М. П. Драгоманова). Можливості роботи з цією програмою детально описано в [5]. У цьому посібнику розглянуто лише ті аспекти роботи із зазначеним програмним засобом, які потрібні для розв'язання задач у контексті матеріалу, що вивчається.

Назва програми GRAN1 продиктована її призначенням — графічний аналіз функцій (*grafic analysis*). Для роботи з програмою потрібен файл GRAN1.exe, а для використання в роботі контекстної допомоги — також файл GRAN1.hlp. Сумарний обсяг пам'яті, потрібної для роботи з програмою, не перевищує 240 кбайт. Після запуску на виконання файла GRAN1.exe на екрані з'являється зображення, подане на рис. 8.1.

Екран умовно поділено на шість частин. Верхню частину займає головне меню, кожна команда якого дає змогу звернутися до певного набору команд (підменю). Центральне місце займає вікно “Графік”, призначене для зображення графіків розглядуваних функцій.

Вікно “Опції” відображує кількість розглядуваних функцій, їх назви, а також тип функціональної залежності. Одночасно можна працювати не більше ніж з п'ятьма функціями. Кожну з них можна задати за допомогою одного з таких типів функціональних залежностей:

- явної залежності $y = y(x)$;
- параметричної з використанням параметра t : $x = x(t)$, $y = y(t)$;
- полярної, заданої в полярних координатах у вигляді $r = r(F)$, де r — полярний радіус точки на площині; F — полярний кут;
- неявної $G(x, y) = 0$;
- табличної.

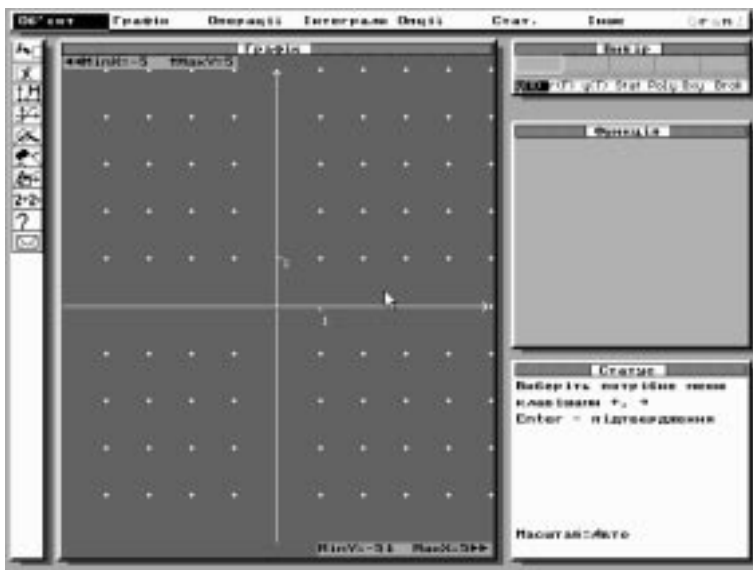


Рис. 8.1

Програмою GRAN1 передбачено можливість роботи зі статистичною вибіркою й об'єктом типу “Ламана”. Для задання типу функціональної залежності використовують команду “Встановити тип” пункту головного меню “Опції” (рис. 8.2). Після зазначення типу функціональної залежності в разі звернення до команди “Новий об'єкт” користувачеві пропонується ввести функцію згідно з визначеним типом.

Приклад 8.1. Розглянемо функцію $3x + 7y = 4$. Перетворимо вираз, що описує функцію, до вигляду $G(x, y) = 0$ (неявне подання функції):

$$3x + 7y - 4 = 0. \quad (8.1)$$

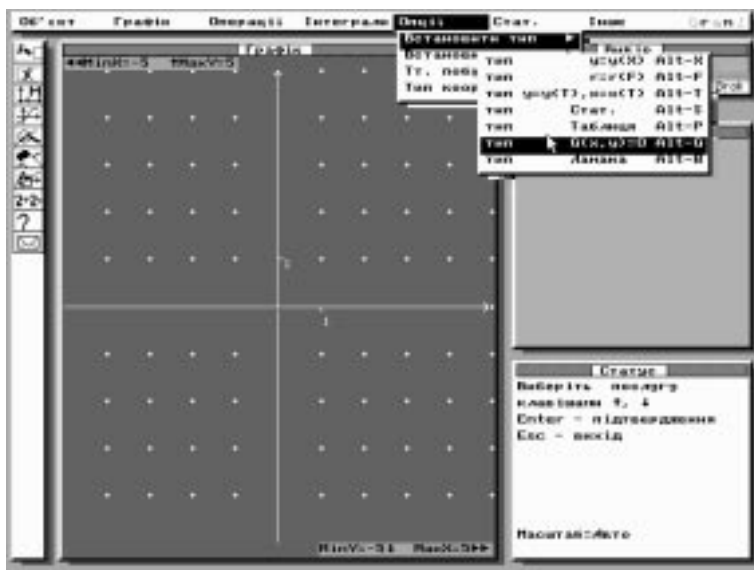


Рис. 8.2

Вибираємо з головного меню пункт “Опції”, далі — пункт “Встановити тип”, а потім — “тип $G(x, y) = 0$ ”. У головному меню послідовно вибираємо пункти “Об’єкт”, “Новий об’єкт”. Після цього послідовно вводимо ліву частину виразу (8.1) за допомогою клавіатури чи панелі калькулятора, запропонованої програмою (рис. 8.3).

Щоб увести вираз за допомогою панелі калькулятора, потрібно помістити курсор на відповідну кнопку калькулятора й натиснути клавішу “Пропуск”. Закінчують вводити вираз вибором на панелі калькулятора кнопки “Введення”. Зручно вводити вирази за допомогою маніпулятора “миша”.

Уведений вираз відображається у вікні “Функція”. Щоб отримати у вікні “Графік” зображення графіка заданої функції, потрібно звернутися до пункту “Графік” головного меню та вибрати підпункт “Побудувати”. Можна також натиснути клавішу F5 (рис. 8.4). У нижньому правому куті екрана є вікно “Статус”, яке містить інформацію про поточний стан роботи з програмою.

У лівій частині екрана є набір піктограм, які дають змогу прискорити роботу з програмою. Поточну дію під час роботи з програмою

GRAN1 можна відмінити за допомогою клавіші Esc, а одні й ті самі функціональні клавіші в різних ситуаціях можуть мати різне призначення.

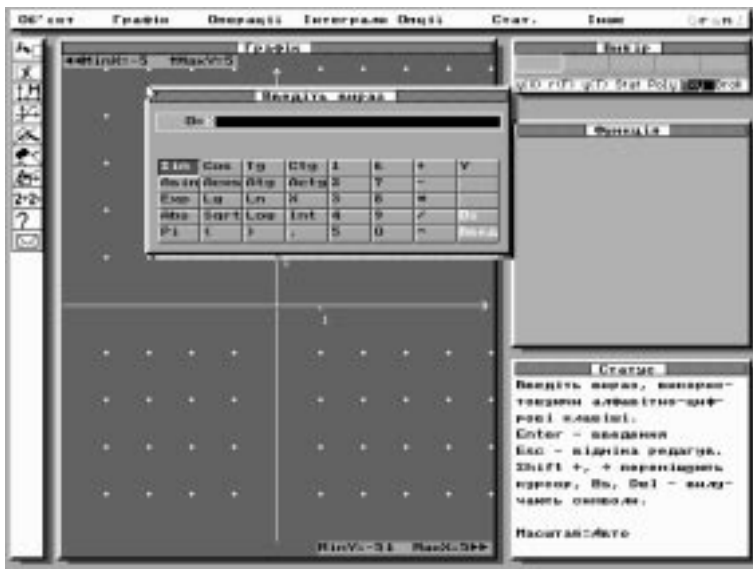


Рис. 8.3

Розглянемо деякі можливості програми та її особливості на прикладах розв'язання задач.

Графічне розв'язання ЗЛП. Можна використовувати програмний засіб GRAN1 для розв'язування ЗЛП в таких випадках:

- 1) якщо обмежень не більше чотирьох;
- 2) змінних в обмеженнях-нерівностях не більше двох.

Задача 8.1. Побудувати багатокутник розв'язків ЗЛП, заданий такими обмеженнями-нерівностями:

$$\begin{cases} 2x + 5y < 10, \\ 5x + 2y < 10, \\ x + y < 2,5, \\ x > 0, y > 0. \end{cases}$$

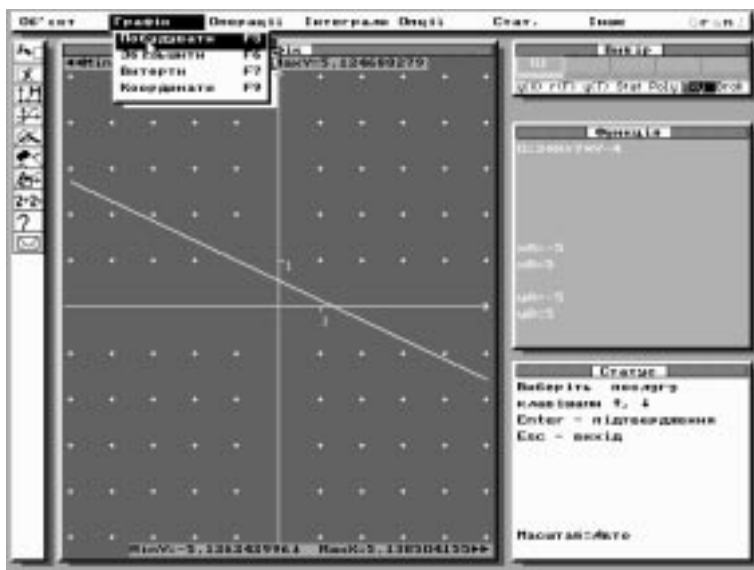


Рис. 8.4

Установимо тип функціональної залежності (неявно задана функція), вибираючи послідовно команди “Опції” — “Встановити тип” — “тип $G(x, y) = 0$ ”.

Запишемо обмеження у вигляді $G(x, y) > 0$:

$$\begin{cases} -2x - 5y + 10 > 0, \\ -5x - 2y + 10 > 0, \\ -x - y + 2,5 > 0, \\ x > 0, y > 0. \end{cases} \quad (8.2)$$

Розглянемо п'ять функцій зазначеного типу, вибираючи послідовно опції “Об’єкт” — “Новий об’єкт”. Вирази в лівій частині обмежень (8.2) вводять безпосередньо з клавіатури чи за допомогою калькулятора на екрані. Після введення кожної з функцій у вікні “Вибір” з’являється позначення для введеної функції, а у вікні “Функція” — введені вирази.

Усі функції мають бути активними. Підтвердження того, що функція активна, — підкреслене її значення у вікні “Вибір”. Якщо

позначення у вікні “Вибір” не підкреслено, тобто функція не активна, її потрібно активізувати:

- вибрати з головного меню опцію “Об’єкт”;
- вибрати опцію “Вибір”;
- помістити курсор у вікні “Вибір” за допомогою стрілок \rightarrow і \leftarrow на позначення функції;
- натиснути клавішу “Пропуск”.

Аналогічно можна відмінити активність функції.

Після того, як уведено до розгляду всі функції, причому всі вони позначені як активні, потрібно побудувати їх графіки. Для цього використовують команди “Графік” — “Побудувати”. У вікні “Графік” з’явиться зображення п’яти графіків (рис. 8.5).

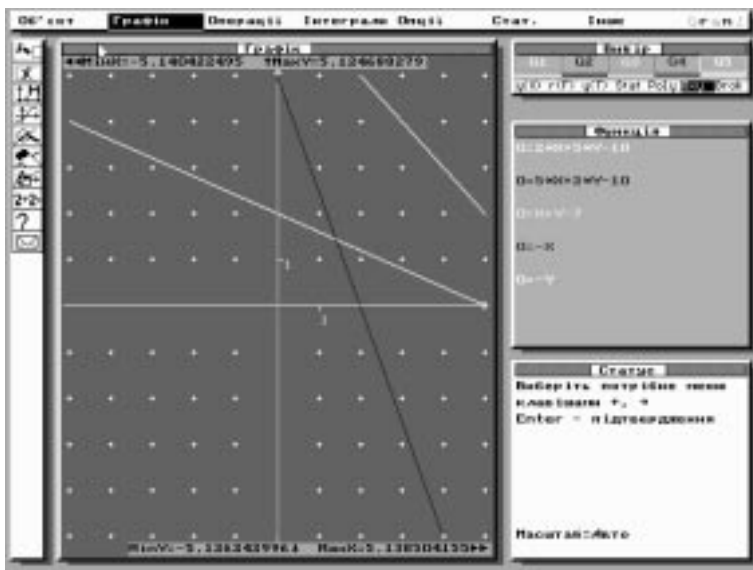


Рис. 8.5

Для подальшого розв’язання потрібно змінити відрізки спостереження на координатних осях за допомогою команд “Опції” — “Встановити масштаб” — “Масштаб корист. Alt-U” (рис. 8.6). На запит системи вводимо $X_A = -1$; $X_B = 6$; $Y_A = -1$; $Y_B = 6$. Для наочності можна використати команди “Об’єкт” — “Змінити відрізок”, оскільки

ки спочатку було встановлено $X_A = Y_A = -5$; $X_B = Y_B = 5$. Потім із головного меню вибираємо пункт “Операції”, а з нього — “Система нерів. $G(x, y) \geq 0$ ”. Після звернення до цієї послуги у вікні “Графік” заштриховується область із точок, координати яких задовольняють нерівності (8.2) (рис. 8.7). Заштрихована область — це багатокутник розв’язків даної задачі.

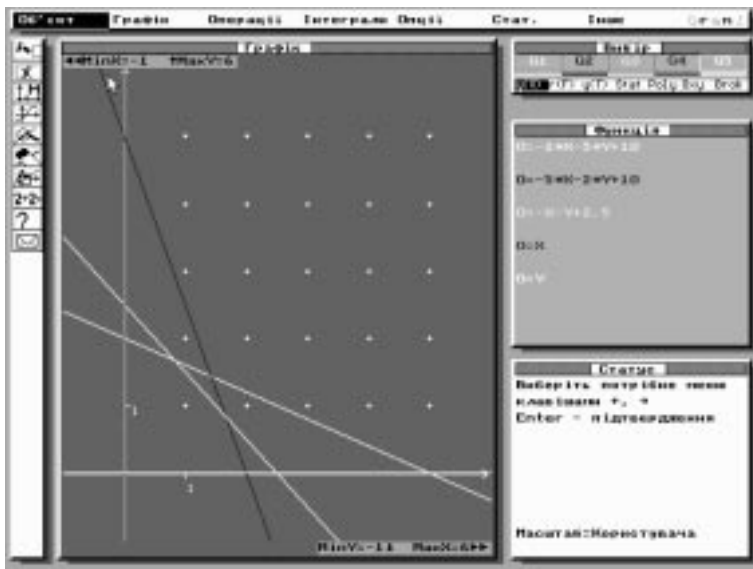


Рис. 8.6

Задача 8.2. Розв’язати графічно таку ЗЛП: знайти максимальне значення лінійної функції $Z = 12x_1 + 16x_2$ для обмежень

$$\begin{cases} 5x_1 + 4x_2 < 20, \\ 16x_1 + 4x_2 < 48, \\ 3x_1 + 8x_2 < 30, \\ x_1 > 0, x_2 > 0. \end{cases}$$

Кількість обмежень-нерівностей не перевищує трьох ($x_1 > 0$, $x_2 > 0$ не враховуємо), а змінних у запису функції дві, тобто можна

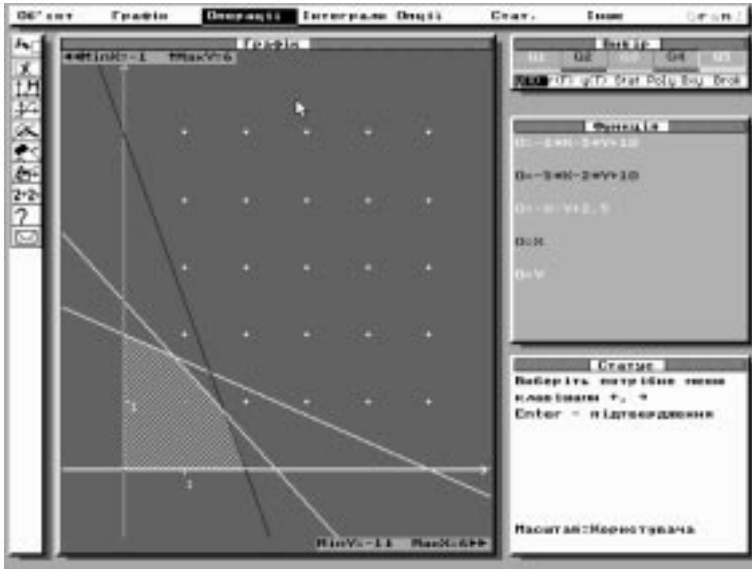


Рис. 8.7

використовувати програму GRAN1 для графічного розв’язування задачі.

Перепишемо обмеження-нерівності у вигляді $G(x, y) > 0$:

$$\begin{cases} 20 - 5x_1 - 4x_2 > 0, \\ 48 - 16x_1 - 4x_2 > 0, \\ 30 - 3x_1 - 8x_2 > 0. \end{cases}$$

Замінімо змінну x_1 на x та x_2 на y і розглянемо функції

$$G1(x, y) = 20 - 5x - 4y,$$

$$G2(x, y) = 48 - 16x - 4y,$$

$$G3(x, y) = 30 - 3x - 8y.$$

Для цього спочатку встановимо потрібний тип функції за допомогою команд “Опції”, “Встановити тип” і “тип $G(x, y) = 0$ ”. Після

того, як тип $G(x, y)$ вибрано, задаємо функції $G1(x, y)$, $G2(x, y)$, $G3(x, y)$, тричі використовуючи послідовність команд “Об’єкт” — “Нова функція”.

На запит системи щодо меж уздовж координатних осей (ХА, ХВ, YA, YB) для цієї задачі доцільно залишити запропоновані системою значення $XA = -5$; $XB = 5$; $YA = -5$; $YB = 5$. Графіки введених функцій побудуємо за допомогою команд “Графік” — “Побудувати” (рис. 8.8).

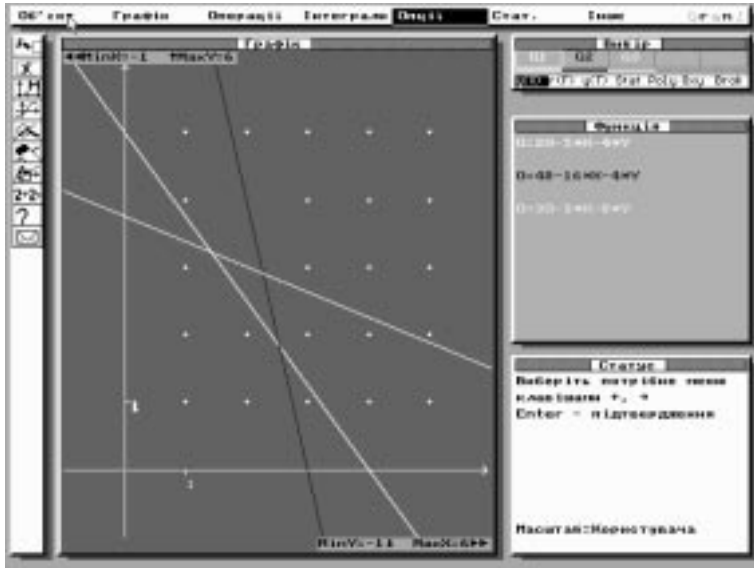


Рис. 8.8

Розглянувши функції $G4 = x$ і $G5 = y$, а потім скориставшись командами “Операції” — “Система нерів. $G(x, y) > 0$ ”, отримаємо багатокутник розв’язків задачі (рис. 8.9).

Для подальшої роботи відкинемо функції $G4$ та $G5$. Для цього потрібно помістити курсор у вікні “Вибір” на одну з функцій і натиснути клавішу F8 (чи “Об’єкт” — “Вилучити”); те саме потрібно повторити для другої функції.

Розглянемо функцію $G4(x, y) = 12x + 16y$, яка відповідає заданій цільовій функції. Отриманий на екрані графік функції $G4$ відповідає значенню цільової функції $Z_0 = 0$, тобто $0 = 12x + 16y$ (рис. 8.10).

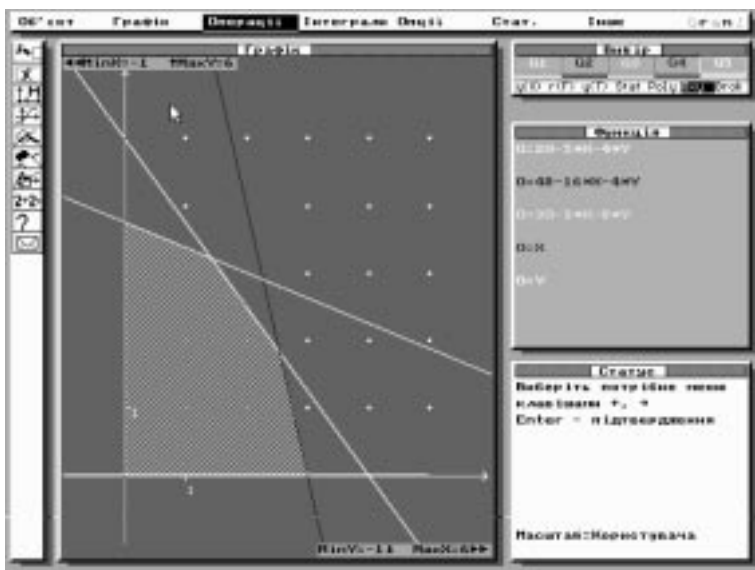


Рис. 8.9

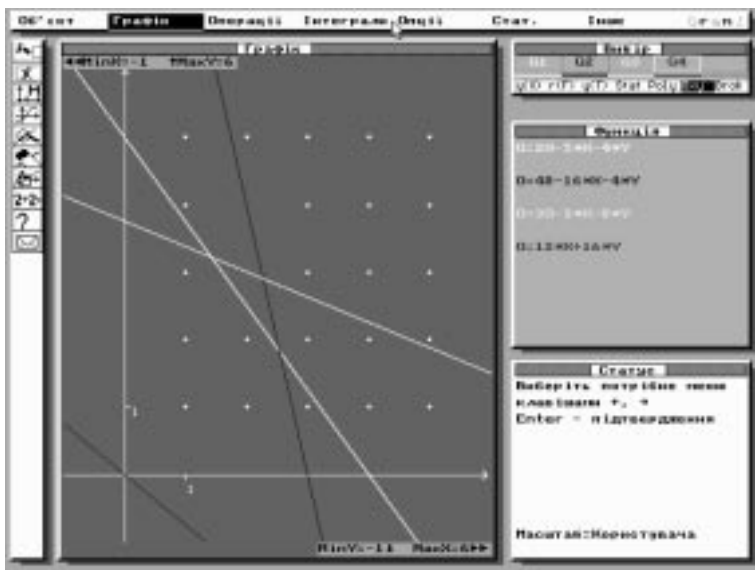


Рис. 8.10

Оскільки потрібно обчислити максимальне значення цільової функції на заданих обмеженнях, то наступний графік потрібно побудувати для функції, що відповідає значенню $Z_1 > Z_0$, наприклад $15 = 12x + 16y$. Для цього за допомогою команди “Об’єкт” — “Замінити функцію” замінимо функцію $G4(x, y)$ функцією $G4(x, y) = 12x + 16y - 15$. Нагадаємо, що курсор у вікні “Вибір” має бути на функції $G4$. Змінивши запис функції $G4$ та проаналізувавши новий графік (рис. 8.11), можна дійти висновку, що значення $Z = 15$ — не оптимальний розв’язок для заданих обмежень.

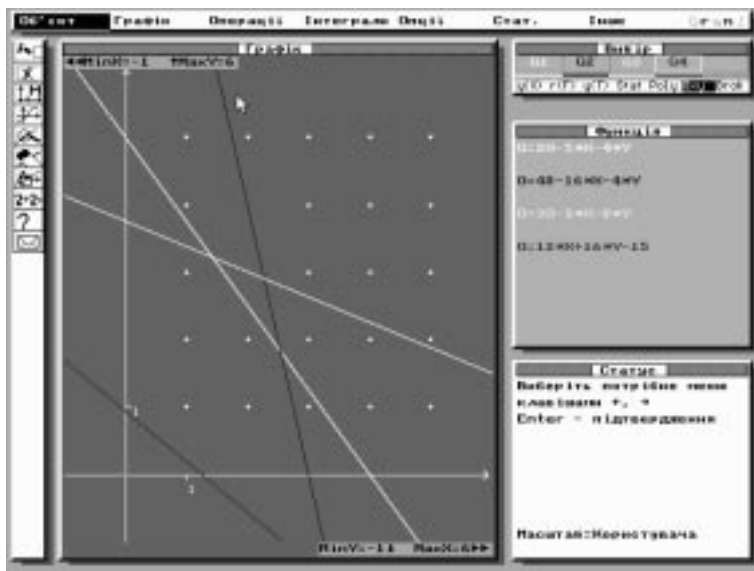


Рис. 8.11

Збільшуючи значення Z , повторюючи процедуру зміни функції та побудови графіка з урахуванням того, що цільова функція може досягати оптимального значення у вершинах багатокутника розв’язків, підберемо таке значення $Z'_{\text{опт}}$, для якого графік функції $G4$ проходить через точку перетину графіків функцій $G1$ і $G3$. Очевидно, що підбране таким способом значення наближене. Для точнішого підбору значення $Z'_{\text{опт}}$ потрібно скористатися командою “Графік” — “Збільшити”. Слід у вікні “Графік” виділити місце для збільшення —

область перетину графіків $G1$ і $G3$ (рис. 8.12), а потім за допомогою команди “Змінити функцію” продовжити підбір значення $Z'_{\text{опт}}$ (рис. 8.13). За допомогою описаного способу отримаємо значення цільової функції $Z'_{\text{опт}} = 68,55$.

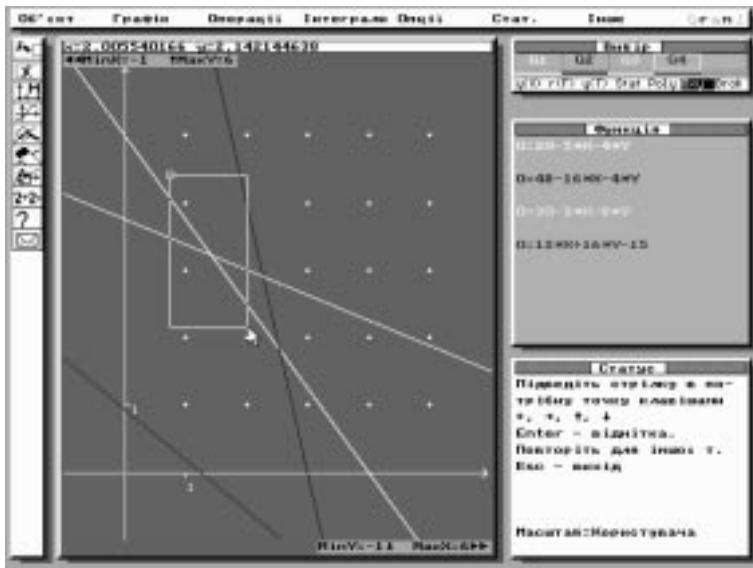


Рис. 8.12

Щоб знайти не тільки оптимальне значення $Z'_{\text{опт}}$, а й оптимальний план, на якому воно досягається, потрібно за допомогою команди “Графік” — “Координати” знайти координати точок перетину графіків функцій $G1$ і $G3$. Точніші значення координат точок перетину можна отримати за допомогою команди “Графік” — “Збільшити”. Тоді оптимальний план $X_{\text{опт}} = (1,42; 3,22)$ (рис. 8.14).

Одержані значення компонент оптимального плану наближені, але за допомогою команд “Збільшити” та “Координати” можна досягти досить високої точності результату.

У ході розв’язання задачі в програмі GRAN1 досить часто потрібен не автоматично запропонований системою масштаб, а так званий “Масштаб користувача”, що дає змогу раціонально розміщувати графіки на екрані. Установити бажані межі вздовж осей координат

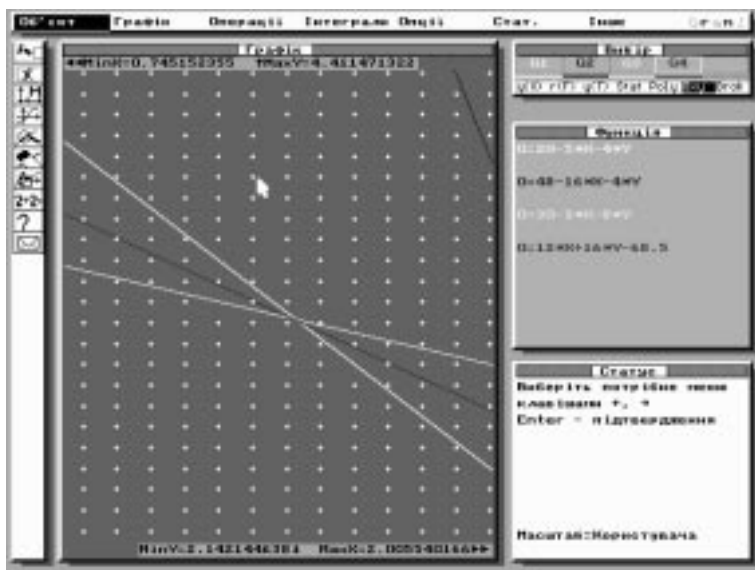


Рис. 8.13

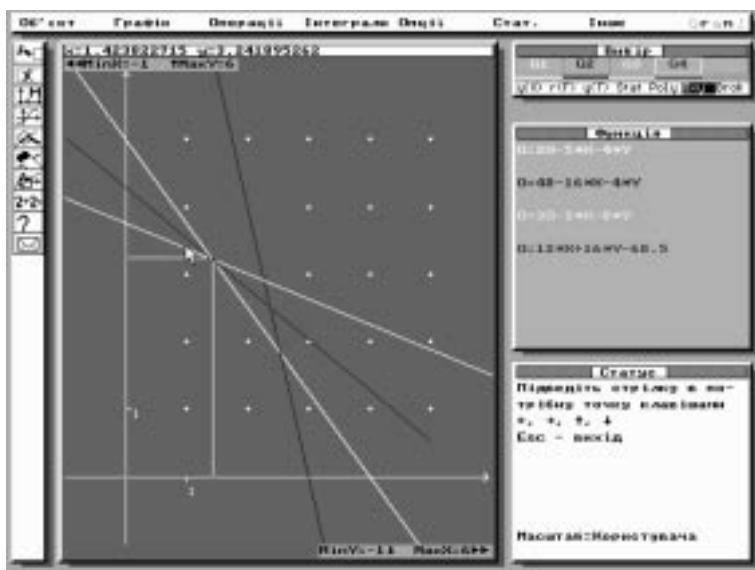


Рис. 8.14

дають змогу команди “Опції” — “Встановити масштаб” — “Масштаб корист.”.

Графічне розв’язування ЗНП.

Задача 8.3. Знайти максимальне та мінімальне значення функції

$$Z = x_1^2 + x_2^2 - 4x_1 - 4x_2 + 8$$

для обмежень

$$\begin{cases} x_1 + x_2 < 7, \\ x_1 > 0, x_2 > 0. \end{cases}$$

Розглянемо функції $G1(x, y) = 7 - x - y$, $G2(x, y) = x$, $G3(x, y) = y$ відповідно до обмежень-нерівностей. Використовуючи відомі команди, отримаємо багатокутник розв’язків даної задачі (рис. 8.15).

Перетворивши вираз для функції Z , легко помітити, що рівняння $Z = (x_1 - 2)^2 + (x_2 - 2)^2$ описує коло радіусом \sqrt{Z} із центром у точці $(2; 2)$.

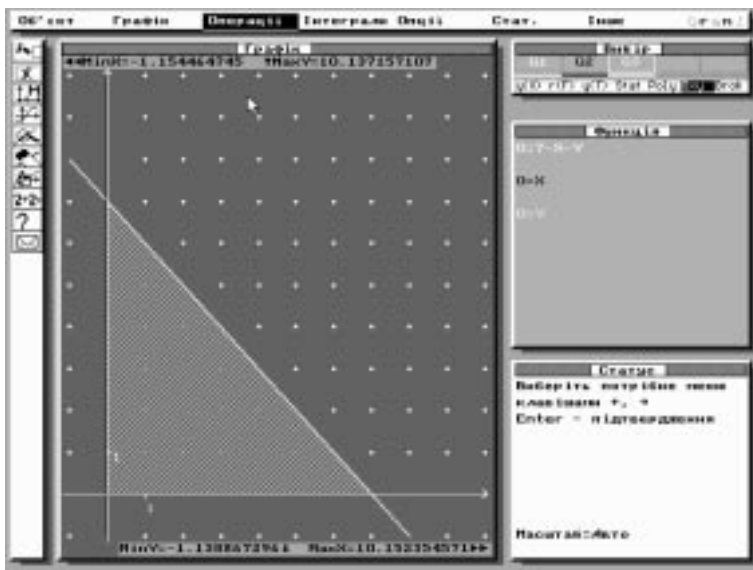


Рис. 8.15

Розглянемо функцію $G4(x, y) = (x - 2)^2 + (y - 2)^2$ та побудуємо її графік — точку з координатами $(2; 2)$. За допомогою команди “Змінити функцію” змінимо запис функції $G4$. Нехай $G4(x, y) = (x - 2)^2 + (y - 2)^2 - 1$. Графік цієї функції — коло радіусом 1 із центром у точці $(2; 2)$ (рис. 8.16). Будемо збільшувати радіус кола доти, доки воно не дотикнеться до прямої. Останній варіант для функції $G4$ пропонуємо такий: $G4(x, y) = (x - 2)^2 + (y - 2)^2 - 4,53$, тобто радіус кола дорівнює $\sqrt{4,53}$, що відповідає значенню цільової функції $Z_{\max} = 4,53$ (рис. 8.17). За допомогою команд “Збільшити” та “Змінити функцію” можна досягти точнішого результату. Очевидно, що для відшукування найменшого значення функції Z не потрібні ніякі побудови, бо воно досягається в точці з координатами $(2; 2)$ області розв’язку й дорівнює нулю $Z_{\min} = 0$.

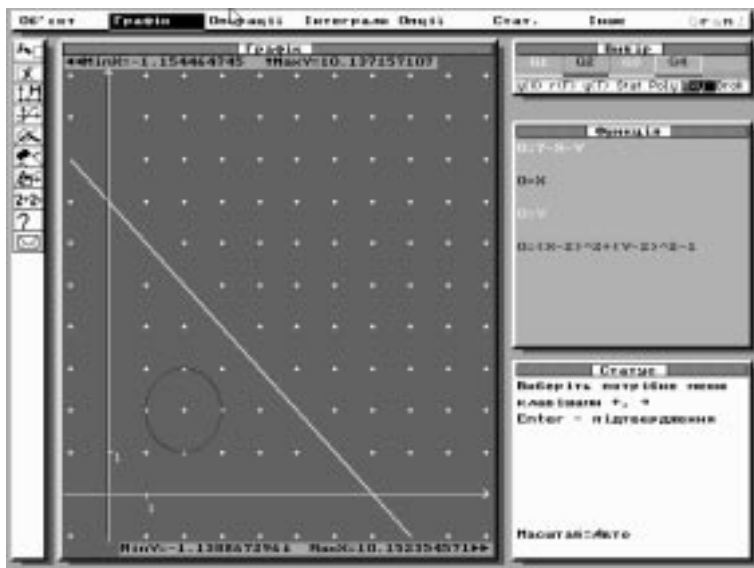


Рис. 8.16

Якщо графік цільової функції — коло з відомим центром, то за допомогою програми GRAN1 можна розв’язати задачу іншим способом. Після того, як багатокутник розв’язків задачі побудовано, із головного меню обирають команду “Операції”, а потім “Відстань до точки” (рис. 8.18). Координати точки $(2; 2)$ можна вводити як із клавіатури,

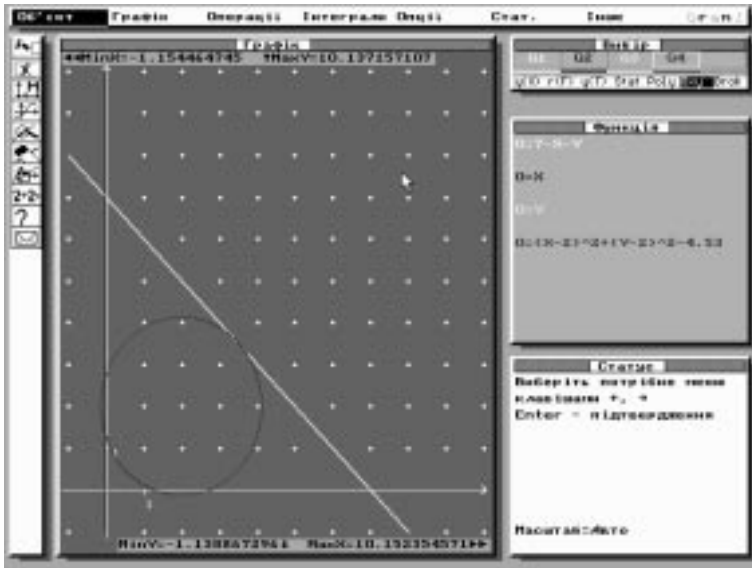


Рис. 8.17

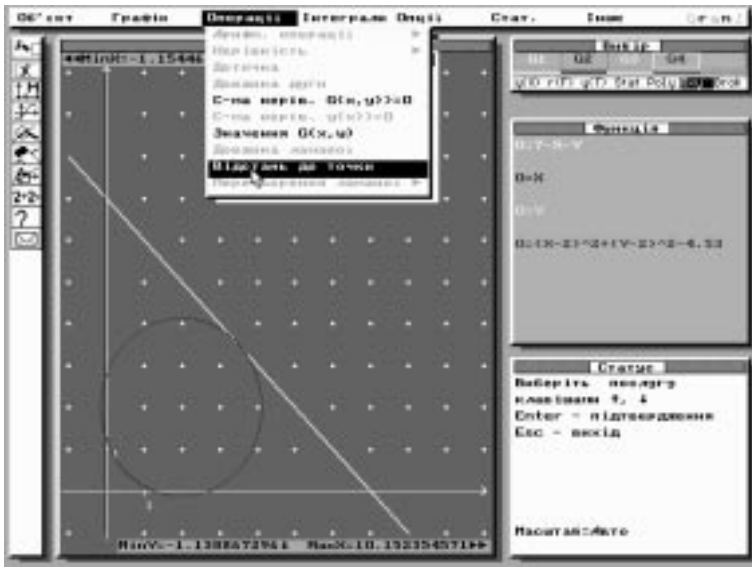


Рис. 8.18

так і за допомогою “миші”. Задану точку позначено на екрані хрестиком; за допомогою “миші” потрібно показати точку, від якої обчислюють відстань до заданої точки (рис. 8.19). Очевидно, що для цієї задачі точку на прямій $G1(x, y) = 7 - x - y$ потрібно відмітити так, щоб пряма $7 - x - y = 0$ та пряма, яка проходить через дві задані точки ((2; 2) та відмічену точку на прямій), були перпендикулярні. У верхній частині вікна “Графік” відображаються координати точки, у якій цільова функція досягає найбільшого значення, і відстань між заданими точками. У цій задачі $x \approx 3,54$; $y \approx 3,47$; $r \approx 2,13$; отже, $Z_{\max} = r^2 \approx 2,13^2 \approx 4,54$. Як бачимо, відповіді, отримані різними способами, майже однакові.

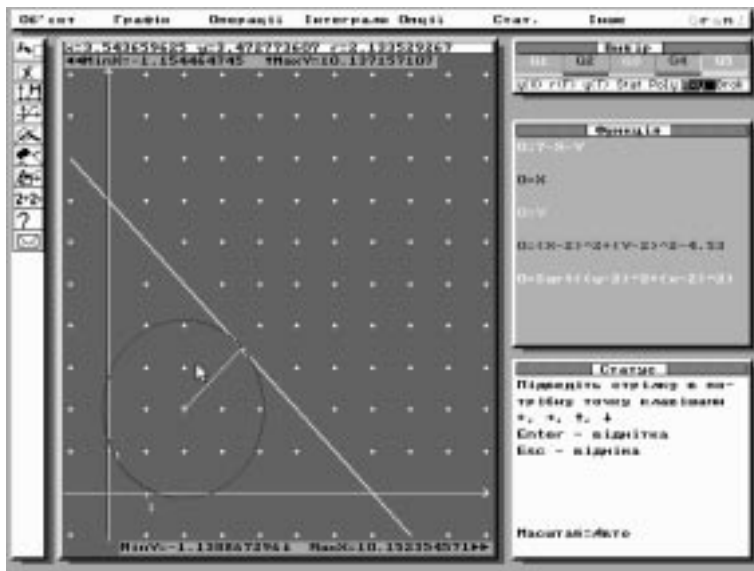


Рис. 8.19

Задача 8.4. Знайти найбільше та найменше значення функції $Z = x_1^2 - 6x_1 + x_2^2 - 6x_2 + 18$ для обмежень

$$\begin{cases} x_1 + x_2 \geq 4, \\ x_1 x_2 \leq 3, \\ 0 \leq x_1 \leq 5, \\ 0 \leq x_2 \leq 6. \end{cases}$$

Позначимо x_1 як x , а x_2 — як y і розглянемо функції $G1(x, y) = x + y + 4$; $G2(x, y) = 3 - xy$; $G3(x, y) = 5 - x$; $G4(x, y) = 6 - y$. За допомогою команд “Опції” — “Система нерів. $G1(x, y) \geq 0$ ” отримаємо область розв’язків цієї задачі (рис. 8.20). Область розв’язків задачі складається з двох частин.

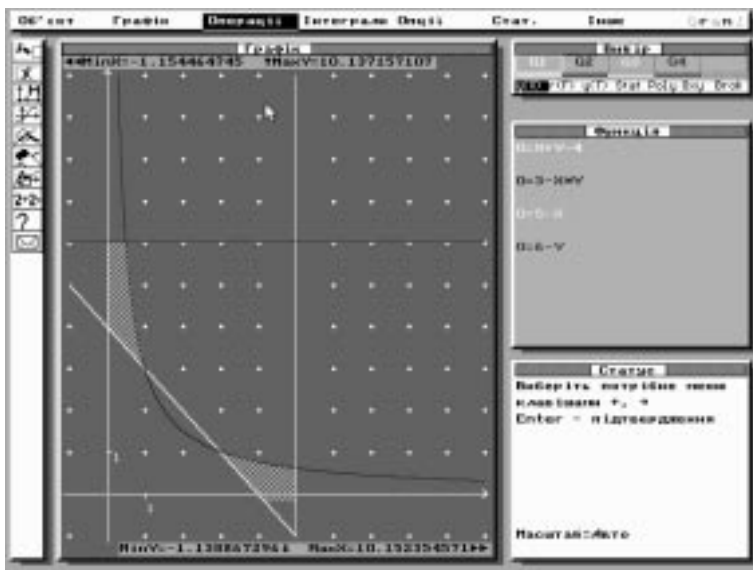


Рис. 8.20

Функцію Z можна подати у вигляді $Z = (x_1 - 3)^2 + (x_2 - 3)^2$. Це рівняння описує коло радіусом \sqrt{Z} із центром у точці $(3; 3)$. Використавши команди “Операції” — “Відстань до точки”, дійдемо висновку, що задача має два локальні мінімуми $Z_{\min 1}$ і $Z_{\min 2}$ в точках відповідно $(3; 1)$ і $(1; 3)$ та два локальні максимуми $Z_{\max 1}$ і $Z_{\max 2}$ в точках із координатами відповідно $(0; 6)$ і $(5; 0)$. При цьому $Z_{\min 1} = Z_{\min 2} = 4$; $Z_{\max 1} = 18$; $Z_{\max 2} = 13$.

Точні значення Z_{\max} і Z_{\min} можна отримати за допомогою команд “Графік” — “Збільшити”. Потрібно розглянути функцію $G5(x, y) = (x - 3)^2 + (y - 3)^2$ та підібрати радіус кола за допомогою команд “Об’єкт” — “Змінити функцію” (рис. 8.21–8.23).

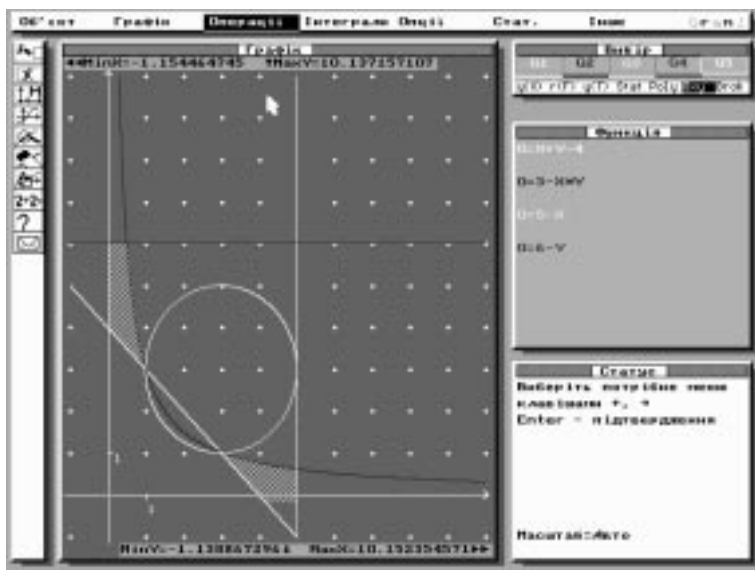


Рис. 8.21

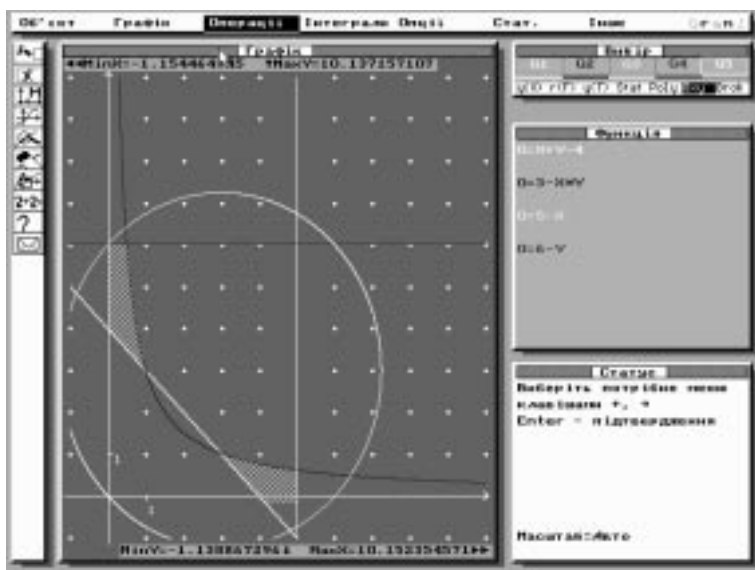


Рис. 8.22

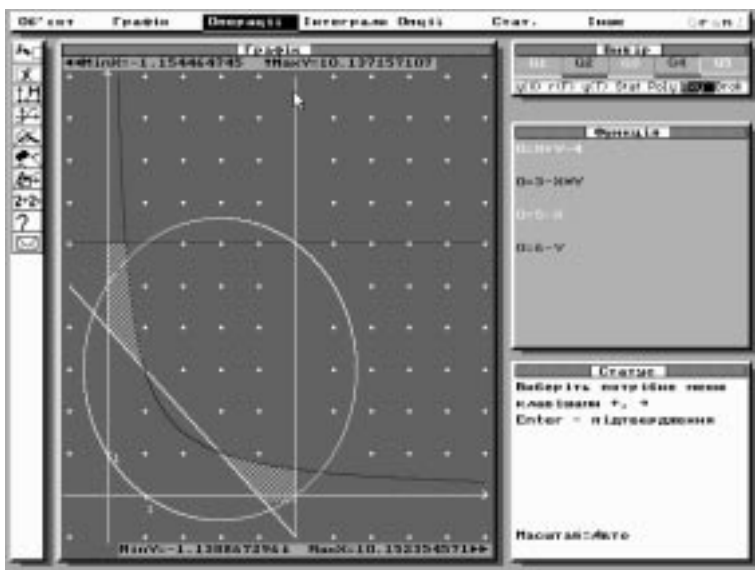


Рис. 8.23

Розв'язування задач із використанням програмного засобу DERIVE

У ході аналітичного розв'язання задач математичного програмування досить часто потрібно виконувати громіздкі обчислення як над числами, так і над масивами чисел (матрицями, векторами тощо). Прискорити виконання обчислювальних операцій і зменшити ймовірність помилок під час розв'язання задач дає змогу програмний засіб DERIVE. Вибір цієї програми не випадковий, бо вона досить відома серед користувачів ПК, повно описана в спеціальній літературі, зручна у використанні, має широкий спектр можливостей.

Програму DERIVE призначено для розв'язування низки математичних задач у символічному вигляді. Це спрощення виразів, виконання арифметичних операцій, розкладання на множники, визначення границь (як у точці, так і на нескінченності), обчислення похідних, інтегралів, розв'язання рівнянь, виконання дій над матрицями (векторами) та ін. Можна також будувати графіки функцій на площині

та зображення поверхонь у тривимірному просторі. Однак графічні можливості програми GRAN1 на площині значно ширші, ніж у програмі DERIVE.

Для роботи з програмою DERIVE (для WINDOWS) потрібно мати файл DFW.exe. Після його активізації на екрані з'являється заставка, подана на рис. 8.24. Спочатку активізується алгебричне вікно, про що свідчить надпис у верхній лівій частині екрана. Можна працювати й у графічних режимах 2D-PLOT Windows, 3D-PLOT Windows у дво- та тривимірному просторі.



Рис. 8.24

У верхній частині екрана розміщене головне меню програми, яке складається з набору команд (стандартного для Windows). Для роботи з головним меню використовують зазвичай маніпулятор “миша”. Для вибору команди з головного меню можна скористатися “гарячою клавішею” — підкресленою літерою в назві команди (наприклад, L для вибору команди soLve) або піктограмою.

Вираз для роботи потрібно ввести з клавіатури, завантажити з файла чи скористатися панеллю з набором символів, запропонува-

ною програмою. Кожен уведений вираз відображається в алгебричному вікні за відповідним номером. Призначення команд головного меню будемо висвітлювати в ході розв'язання задач.

Задача 8.5. Для функції $F(x, y, z, k, m)$ знайти частинні похідні $\frac{\partial F}{\partial x}$, $\frac{\partial F}{\partial y}$, $\frac{\partial F}{\partial z}$, $\frac{\partial F}{\partial k}$, $\frac{\partial F}{\partial m}$, якщо $F = xy + yz + k(x + y - 2) + m(y + 2)$.

Щоб розглянути функцію F , із головного меню виберемо команду Author (клацнемо правою клавішею “миші” на слові Author). Із додаткового меню вибираємо Expression... і далі вводимо з клавіатури вираз для функції F , а потім із головного меню вибираємо команду Calculus. Після звернення до послуги Calculus на місці головного меню з'являється підменю вибраної команди:

Limit ...
 Differentiate ...
 Taylor series ...
 Integrate ...
 Sum ...
 Product ...
 Vector ...

Для обчислення похідних використовуємо команду Differentiate, після вибору якої з'являється запит виразу, який потрібно диференціювати, змінної, за якою потрібно диференціювати, і порядку похідної.

У відповідь на запит потрібно ввести номер виразу, під яким записано функцію (1), змінну диференціювання variable (x) та порядок похідної order (1). Після цього під номером 2 в алгебричному вікні з'явиться запис

$$\frac{d}{dx}(xy + yz + k(x + y - 2) + m(y + z - 2)).$$

Далі вибираємо команду головного меню Simplify, а з поданого підменю вибираємо команду Expand. Після звернення до команди Expand на екрані з'явиться вікно — запит відносно номера виразу, до якого потрібно застосувати цю команду; змінних, використаних у вибраному виразі, та вигляду виразу після виконання дій (Trivial, SquareFree, Rational, Redical).

Відповівши на всі запити у вікні запитів, потрібно обрати команду Expand.

Під номером 3 в алгебричному вікні записується результат диференціювання функції F за змінною x , тобто $k + y$. Аналогічно знаходять інші частинні похідні (рис. 8.25).



Рис. 8.25

Якщо, відповідаючи на запит щодо змінної диференціювання та порядку похідної, натиснути кнопку "Simplify", а не "OK", то в алгебричному вікні буде записано відразу результат виконання диференціювання. На рис. 8.25 в алгебричному вікні під номерами 8 і 9 записано результати диференціювання функції F відповідно за змінними k й m .

Зауваження. За допомогою команди Calculus програми DERIVE можна визначати границі (одно- та двосторонні) й інтеграли (невизначені та визначені), розвивати функцію в ряд Тейлора із заданою

кількістю перших членів розвинення, обчислювати суми типу $\sum_{i=a}^k S$ (S — якийсь вираз; i — параметр суми; a, k — відповідно нижня та

верхня межі додавання) та добутки типу $\prod_{i=a}^k P$ (рис. 8.26).

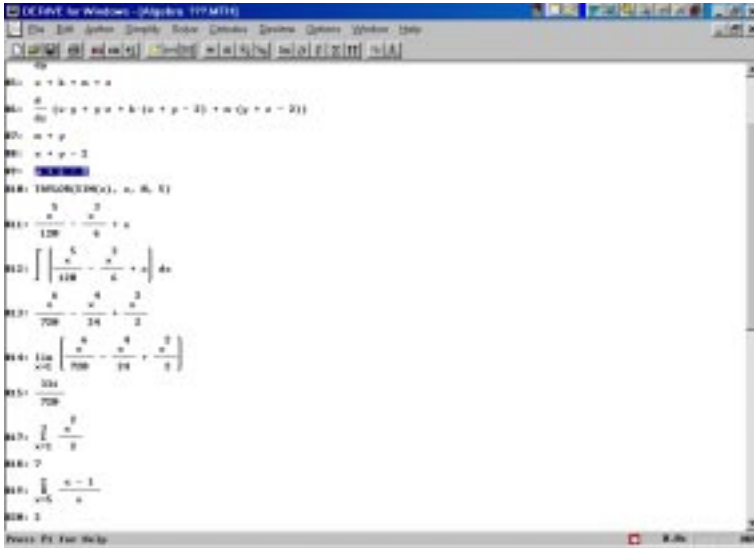


Рис. 8.26

Задача 8.6. Розв'язати систему рівнянь, використовуючи обернену матрицю:

$$\begin{cases} x_1 + 2x_2 - x_3 = 4, \\ 2x_1 - x_2 + x_3 = 12, \\ 3x_1 + 2x_2 + 3x_3 = 26. \end{cases}$$

Розв'язок цієї системи — матриця-стовпець

$$X = A^{-1}B,$$

де $X = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$ — стовпець невідомих; A^{-1} — матриця, обернена до мат-

риці $A = \begin{pmatrix} 1 & 2 & -1 \\ 2 & -1 & 1 \\ 3 & 2 & 3 \end{pmatrix}$; $B = \begin{pmatrix} 4 \\ 12 \\ 26 \end{pmatrix}$.

Щоб розглянути матриці A та B , використаємо команду головного меню Author-Matrix. Після введення кількості рядків (Rows) і стовпців (Column) з'являється запит щодо елементів матриці. У відповідь потрібно ввести задану кількість виразів (чисел), які визначають елементи матриці. Після цього в алгебричному вікні з'явиться під відповідним номером (у цьому разі під номером 1) матриця A . Аналогічно можна ввести матрицю-стовпець B (рис. 8.27).

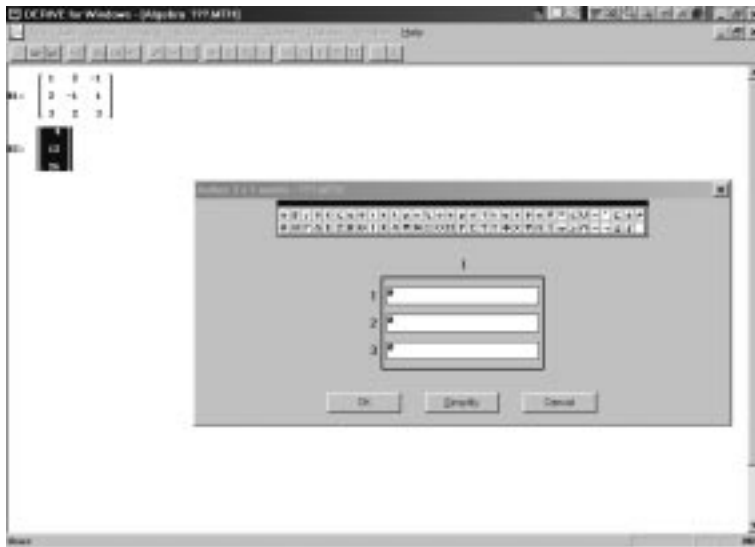


Рис. 8.27

Далі потрібно знайти матрицю A^{-1} , обернену до матриці A . Для цього з головного меню вибираємо команду Auto-Expression. У робочому рядку вікна Auto-Expression уводимо вираз $(\# 1)^{-1}$, де $(\# 1)$ — номер виразу, який відповідає матриці A . Закінчуємо введення вибором команди Simplify. Під номером 3 в алгебричному вікні записано матрицю A^{-1} . Потім за допомогою команди Auto-Expression перемножимо матриці A^{-1} та B . Для цього в робочому рядку вікна Auto-Expression уведемо $(\# 3) \cdot (\# 2)$. Для наочності розв'язання закінчимо введення виразу $A^{-1}B$ натисканням кнопки “OK”. Тоді в алгебричному вікні під номером 4 запишеться множення

матриць A^{-1} та B . Натиснувши ж після введення виразу $A^{-1}B$ кнопку “Simplify”, можна відразу отримати під номером 4 результат множення (рис. 8.28).

Програма DERIVE дає змогу транспонувати матриці, обчислювати суму та різницю матриць із однаковою розмірністю, обчислювати визначник квадратної матриці тощо (рис. 8.29).

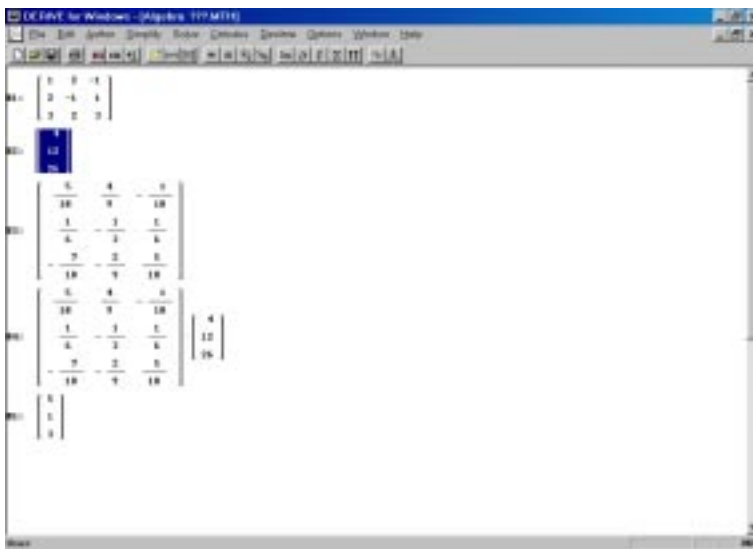


Рис. 8.28

Розв’язування задач із використанням електронних таблиць Excel

Для розв’язання багатьох задач математичного програмування потрібно виконувати “рутинні” та громіздкі обчислення. Інколи для одержання результату треба багато разів провести обчислення за однією й тією самою формулою, замінюючи лише значення деяких її елементів. Збільшити швидкість розв’язування задачі та при цьому не припуститися помилки дає змогу використання відомих і широко розповсюджених електронних таблиць Excel. Ми не будемо описувати загальні можливості Excel, бо вони відомі читачеві, а от спе-



Рис. 8.29

цифічні можливості, знання яких потрібні для розв’язання задач, будемо висвітлювати під час розв’язування.

Використовувати електронні таблиці під час розв’язування задач можна декількома способами:

- 1) передати комп’ютеру лише обчислювальну функцію, тобто в клітинки таблиці “вручну” вводити формули, що містяться в алгоритмі розв’язування задачі;
- 2) максимально використовувати стандартні “вбудовані” функції таблиць Excel.

Розглянемо на конкретних прикладах деякі можливості використання таблиць Excel для розв’язання задач математичного програмування.

Задача 8.7. Знайти матрицю, обернену до заданої матриці A . Розв’язування провести методом зведених жорданових виключень (ЗЖВ) та з використанням стандартних функцій Excel;

$$A = \begin{pmatrix} 5 & 2 & 6 \\ 1 & 0 & 1 \\ 2 & 3 & 1 \end{pmatrix}.$$

Перший спосіб. Розв’яжемо задачу, передаючи Excel лише обчислювальні функції, тобто формули обчислення визначника матриці, а також формули для виконання кроку жорданових виключень будемо заносити до відповідних клітинок таблиці “вручну”.

Визначник Δ матриці A обчислюють за такою формулою:

$$\Delta = a_{11}a_{22}a_{33} + a_{21}a_{32}a_{13} + a_{12}a_{23}a_{31} - a_{31}a_{22}a_{13} - a_{32}a_{23}a_{11} - a_{21}a_{12}a_{33}.$$

Правило виконання одного кроку ЗЖВ із розв’язувальним елементом a_{ij} (стовпець i рядок, яким належить розв’язувальний елемент, називають відповідно розв’язувальними стовпцем і рядком):

- 1) розв’язувальний елемент замінюють одиницею;
- 2) усі інші елементи розв’язувального стовпця залишають без змін;
- 3) усі елементи розв’язувального рядка замінюють на протилежні;
- 4) елементи таблиці, які не належать розв’язувальним рядку та стовпцю, обчислюють за правилом прямокутника (рис. 8.30);
- 5) усі елементи нової таблиці ділять на розв’язувальний елемент.

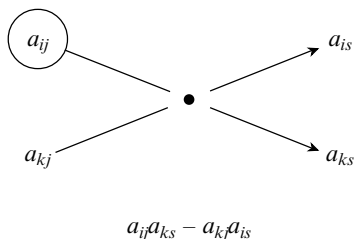


Рис. 8.30

Розмістимо задану матрицю в діапазоні A4 : B6, формулу для обчислення визначника — у клітинку H4, вислідну таблицю після виконання першого кроку ЗЖВ — у діапазоні A10 : C12.

Розглянемо, як формується таблиця в діапазоні A10 : C12. Як розв’язувальний беремо елемент, розміщений у клітинці A4. До клітинки A10 заносимо 1/\$A\$4 (відповідно до першого пункту правила

розв'язувальний елемент замінюємо одиницею, відповідно до п'ятого — усі елементи нової таблиці ділимо на розв'язувальний елемент); до клітинки A11 записуємо формулу $A5/SA\$4$ (відповідно до другого пункту правила усі інші елементи розв'язувального стовпця залишаємо без змін та відповідно до п'ятого пункту — усі елементи нової таблиці ділимо на розв'язувальний елемент).

Для введення формули до будь-якої клітинки потрібно поставити курсор на цю клітинку та розпочати введення знаком “=” (дорівнює) — усе, що буде записано після знака “=”, система Excel сприймає як формулу. Пояснимо також, для чого біля позначення клітинки A4 поставлено знаки “\$”. За їх допомогою розрізняють абсолютну та відносну адреси. Якщо після введення в Excel адреси A4 натиснути один раз клавішу F4, то запис A4 перетвориться на $SA\$4$ — абсолютну адресу A4, що дасть змогу в подальшому в разі копіювання формули, до якої входить $SA\$4$, залишити цю адресу незмінною (відомо, що в разі копіювання формул адреси, які входять до їх складу, змінюються, якщо ці адреси записано у звичайному вигляді, без використання знака “\$”). До клітинки A12 копіюємо формулу з клітинки A11. Оскільки в клітинці A11 записано формулу $A5/SA\$4$, то під час копіювання відносна адреса A5 (записана без знаків “\$”) змінюється на відносну адресу A6 (зсув на один рядок униз), а от абсолютна адреса $SA\$4$ копіюється без змін, тобто в клітинці A12 буде записано формулу $A6/SA\$4$. Якщо поставити знак “\$” тільки перед назвою стовпця (рядка), то під час копіювання незмінним залишиться стовець (рядок). Щоб знак “\$” був тільки перед назвою стовпця, потрібно після введення адреси клітинки натиснути клавішу F4 двічі, а для того, щоб знак “\$” був розміщений тільки перед номером рядка, клавішу F4 потрібно натиснути тричі поспіль. До клітинки B10 записуємо формулу $B4/SA\$4$, яку потім копіюємо до клітинки C10. Оскільки адреса B4 відносна, а $SA\$4$ — абсолютна, то в клітинку C10 під час копіювання записується формула $C4/SA\$4$. Після закінчення дій із розв'язувальними стовпцем і рядком потрібно записати формули для обчислення елементів, які не належать до цих стовпця та рядка. До клітинки B11 записуємо формулу $(SA\$4*B5-SA5*B\$4)/SA\$4$, за допомогою якої відбудеться переобчислення елемента B5 за правилом прямокутника (відповідно до четвертого пункту правила елементи таблиці, які не належать розв'язувальним рядку та стовпцю, обчислюємо за правилом прямокутника та відповідно до п'ятого

пункту усі елементи нової таблиці ділимо на розв'язувальний елемент).

Зазначену формулу вводимо з урахуванням того, що надалі потрібно буде копіювати її в інші клітинки нової таблиці — ті, де розміщено елементи, що не належать розв'язувальним рядку та стовпцю. Саме тому адреса B5 відносна, а \$A\$4 абсолютна. Щодо адрес \$A5 і B\$4 знак “\$” дає змогу зберегти під час копіювання відповідно назву стовпця A та номер четвертого рядка. Після виконання описаних дій до клітинок C11, B12 і C12 буде записано відповідно формули $(\$A\$4*C5-\$A5*C\$4)/\$A\4 , $(\$A\$4*B6-\$A6*B\$4)/\$A\4 та $(\$A\$4*C6-\$A6*C\$4)/\$A\4 .

Другий крок ЗЖВ виконуємо, узявши як розв'язувальний елемент із клітинки B11. Послідовність виконання дій із формування нової таблиці аналогічна попередній. Третій крок ЗЖВ виконуємо з розв'язувальним елементом із клітинки C18.

Правильність одержаного результату — матриці, оберненої до матриці A , — перевіримо, перемноживши вихідну (розміщену в діапазоні A29 : C31) й обернену (розміщену в діапазоні E22 : G31) матриці. Оскільки за домовленістю ми використовуємо табличний процесор Excel без залучення вбудованих функцій, то до клітинок діапазону I29 : K31 запишемо формули для обчислення елементів матриці, що являє собою результат множення вихідної й оберненої матриць: до клітинки I29 запишемо формулу $A29*E29+B29*E30+C29*E31$, до клітинки I30 — формулу $A30*E29+B30*E30+C30*E31$ і т. д. Як бачимо, результат множення — одинична матриця, і це підтверджує правильність одержаного розв'язку.

Надалі, якщо розмістити в діапазоні A4 : C6 будь-яку невиврожену матрицю, то в діапазоні A22 : C24 буде записано обернену до неї. Нагадаємо, що невивроженою називають матрицю, визначник якої відмінний від нуля. У цій задачі визначник матриці обчислено за формулою $A4*B5*C6+A5*B6*C4+B4*C5*A6-A6*B5*C4-B6*C5*A4-A5*B4*C6$, записаною в клітинці H4, а висновок щодо існування оберненої матриці зроблено за формулою ЕСЛИ(H4 = 0; “оберненої матриці не існує”; “обернена матриця існує”), записаною в клітинку E6.

Копії екранів, пов'язаних із розв'язуванням цієї задачі в таблицях Excel, подано на рис. 8.31, 8.32. Перший із них відображає значення клітинок, а другий — формули, які занесено до клітинок.

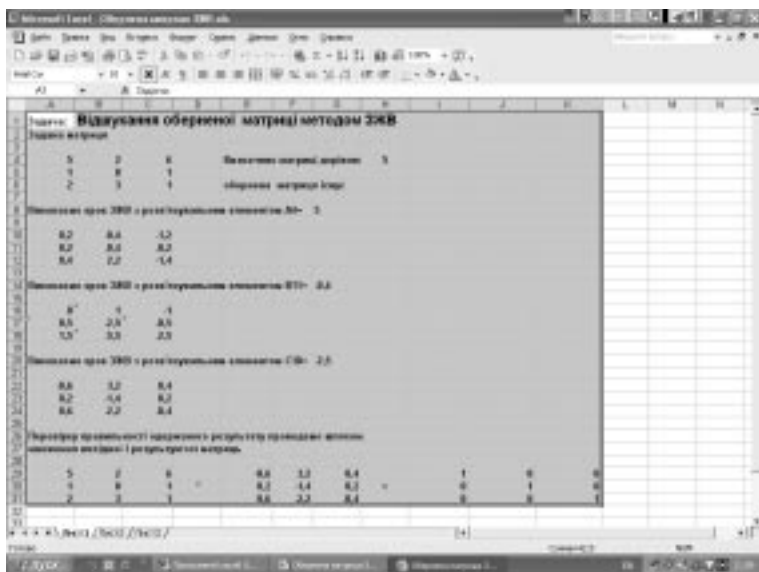


Рис. 8.31

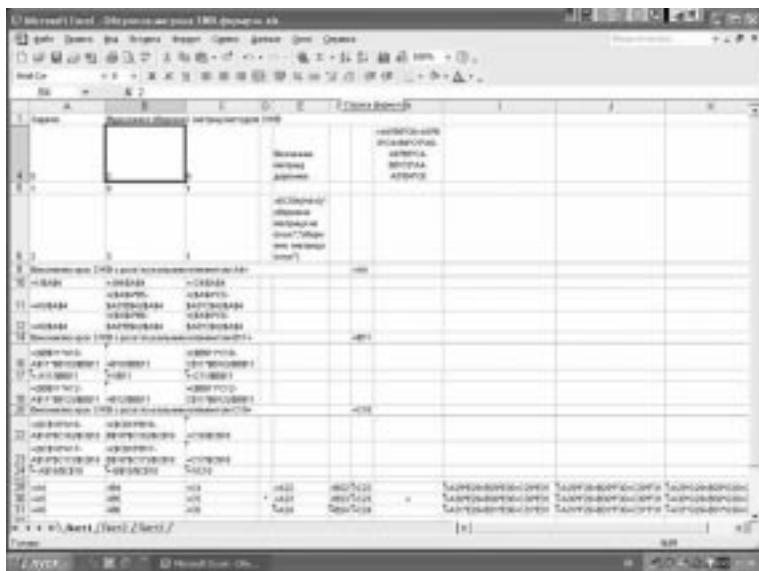


Рис. 8.32

Другий спосіб. Розв’яжемо цю задачу з використанням вбудованих стандартних функцій середовища Excel.

Для вихідної матриці A відводимо діапазон $A4 : C6$. До клітинки $H4$ заносимо формулу $\text{МОПРЕД}(A4 : C6)$, яка повертає значення визначника вихідної матриці (рис. 8.33), до клітинки $E6$ — формулу $\text{ЕСЛИ}(H4=0; \text{“оберненої матриці не існує”}; \text{“обернена матриця існує”})$. Для розміщення вислідної матриці — оберненої — виділяємо діапазон $A10 : C12$, обираємо функцію МОБР , на запит системи вводимо діапазон $A4 : C6$, уведення закінчуємо одночасним натисненням клавіш $\text{Shift, Ctrl, Enter}$ (рис. 8.34). Зображення копії екрана з одержаними результатами подано на рис.35. На рис. 8.36 відображено вміст клітинок таблиці у вигляді формул, за якими одержано значення, подані на рис. 8.35.

Задача 8.8. Розв’язати систему лінійних рівнянь методом ЗЖВ:

$$\begin{cases} 5x_1 + 2x_2 + 6x_3 = 2, \\ x_1 + x_3 = 5, \\ 2x_1 + 3x_2 + x_3 = 1. \end{cases}$$

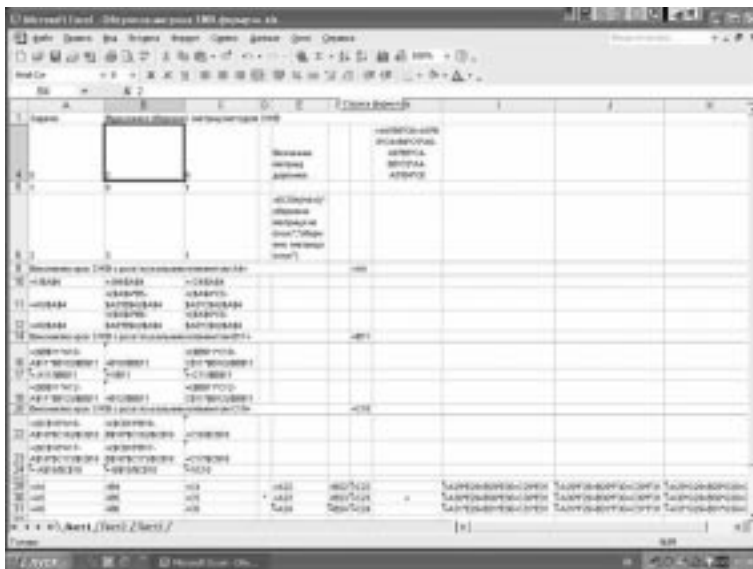


Рис. 8.33

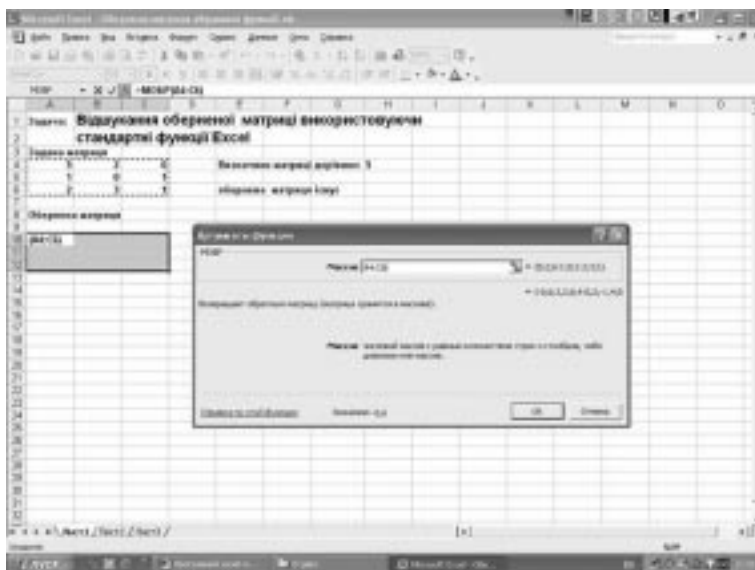


Рис. 8.34

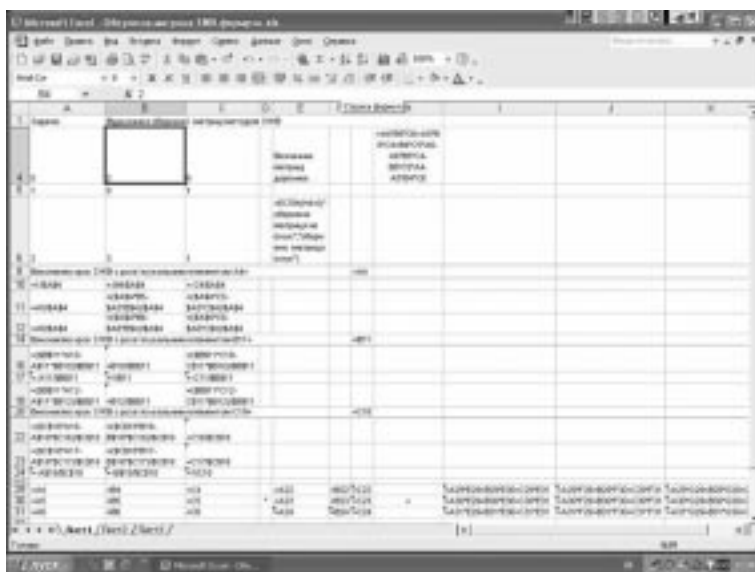


Рис. 8.35

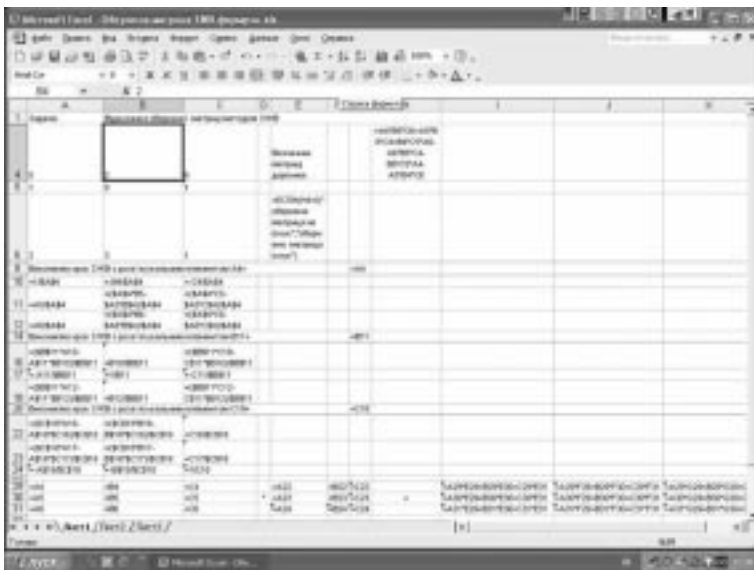


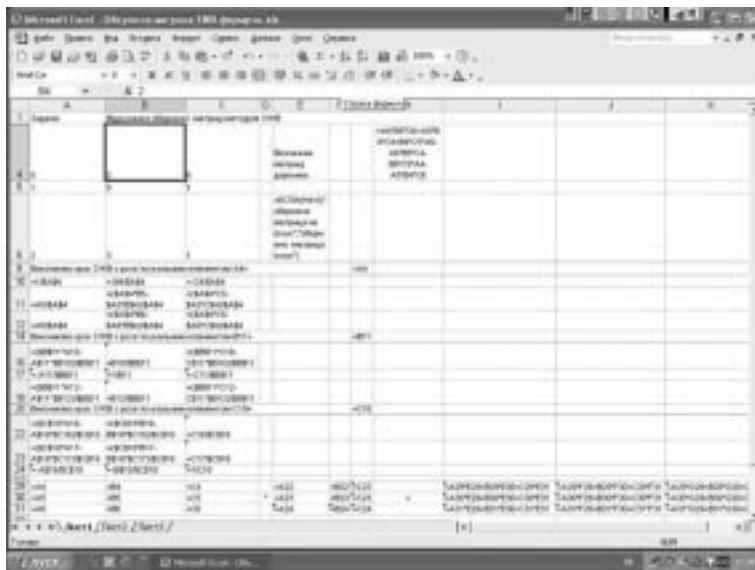
Рис. 8.36

Здану систему рівнянь легко розв'язати розглянутим вище методом ЗЖВ. Для цього перепишемо її у вигляді

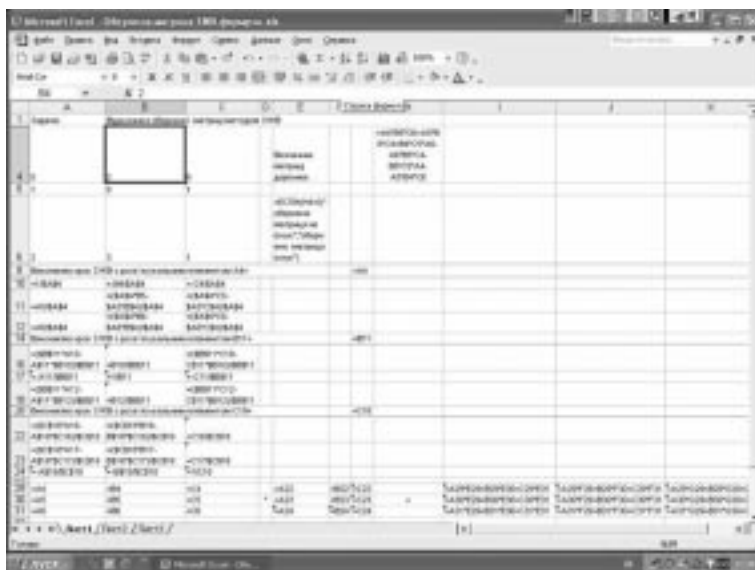
$$\begin{cases} 5x_1 + 2x_2 + 6x_3 - 2 = 0, \\ x_1 + x_3 - 5 = 0, \\ 2x_1 + 3x_2 + x_3 - 1 = 0. \end{cases}$$

та побудуємо жорданову таблицю, останній стовпець якої — стовпець вільних членів зі зміненими на протилежні знаками. Алгоритм подальшого розв'язання нічим не відрізняється від алгоритму визначення оберненої матриці методом ЗЖВ. У вислідній таблиці замість стовпця вільних членів буде розміщено стовпець — розв'язок заданої системи лінійних рівнянь (рис. 8.37). На рис. 8.38 відображено вміст клітинок таблиці (у вигляді формул), за якими одержано значення, подані на рис. 8.37.

Поясимо, як заповнюються клітинки, що містять вільні члени рівнянь заданої системи; усі інші клітинки заповнюються так само, як у викладеному раніше способі (задачі на визначення оберненої



Puc. 8.37



Puc. 8.38

матриці). Під час виконання першого кроку ЗЖВ вільний член першого рівняння, розміщений у клітинці D5, записується в клітинку D11 як результат обчислення за формулою $-D5/SA\$5$. Нагадаємо, що знак “\$” перед назвами стовпця та рядка розв’язувального елемента (у цьому разі це елемент клітинки A5) залишає елемент незмінним під час копіювання в Excel. Вільний член другого рівняння після обчислення за формулою $(SA\$5*D6-DA6*DS5)/SA\5 записуємо до клітинки D12. Формулу до клітинки D12 уводимо копіюванням вмісту клітинки B12. Ситуація з “новим” вільним членом третього рівняння аналогічна: він розміщується в клітинці D13 і обчислюється за формулою $(SA\$5*D7-DA7*DS5)/SA\5 . Легко помітити, що формули для обчислення “нових” вільних членів другого та третього рівнянь відрізняються лише номерами рядків, біля яких не поставлено знак “\$”, а це, як було зазначено раніше, дає змогу заповнити клітинку D13, копіюючи вміст клітинки D12. Номер рядка 6 автоматично замінюється на номер 7. Другий і третій кроки ЗЖВ виконуються аналогічно.

Задача 8.9. Використовуючи стандартні функції Excel, розв’язати систему лінійних рівнянь за допомогою оберненої матриці:

$$\begin{cases} 5x_1 + 2x_2 + 6x_3 - 2 = 0, \\ x_1 + x_3 - 5 = 0, \\ 2x_1 + 3x_2 + x_3 - 1 = 0. \end{cases}$$

Як відомо, для розв’язання поставленої задачі потрібно виконати такі дії:

- визначити матрицю, обернену (якщо вона існує) до основної матриці заданої системи рівнянь;
- перемножити знайдену обернену матрицю та стовпець вільних членів.

Визначення оберненої матриці за допомогою стандартної функції МОБР було розглянуто вище, а щодо використання стандартної функції МУМНОЖ, призначеної для множення матриць, зробимо декілька зауважень. У діапазоні A20 : C22 розмістимо матрицю, обернену до основної матриці заданої системи, у діапазоні E20 : E22 — стовпець вільних членів системи рівнянь; виділимо (за допомогою “миші” чи стрілок і клавіші Shift) діапазон G20 : G22 та звернемося до послуги головного меню “Вставка”. У цьому меню обираємо пункт “Функція...”, далі, визначившись із категорією (“Матема-

тичні”), обираємо із запропонованого переліку функцію МУМНОЖ. На запит системи щодо розміщення матриць-множників вводимо відповідно A20 : C22 та E20 : E22. Для одержання у виділеному діапазоні G20 : G22 результату множення матриць потрібно роботу із запитом функції МУМНОЖ закінчити одночасним натисненням клавіш Shift, Ctrl, Enter (рис. 8.39, 8.40).

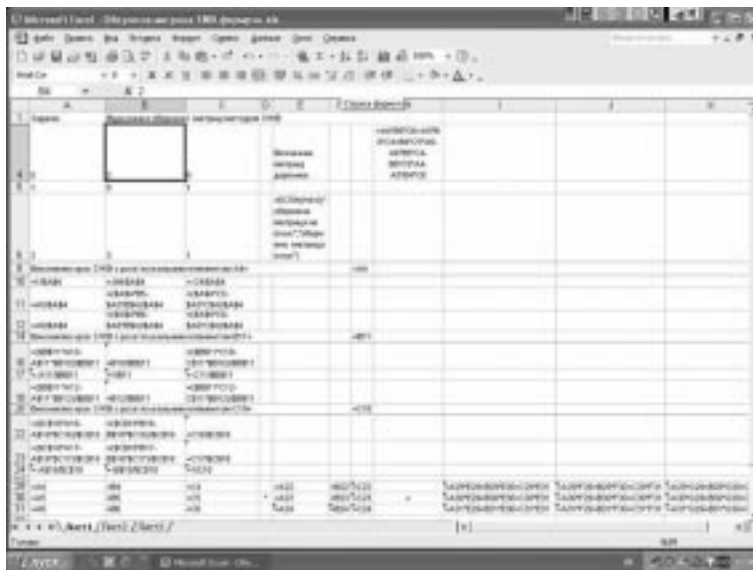


Рис. 8.39

На рис. 8.41 відображено результати розв’язання поставленої задачі, а на рис. 8.42 — вміст клітинок таблиці, але у вигляді формул, за якими були одержані проміжні та вислідні значення.

Задача 8.10. У таблиці містяться показники виконання балансу за звітний період (в умовних грошових одиницях).

Галузь	Використання		Обсяг	
	1	2	кінцевого продукту	валового випуску
1	120	180	240	540
2	280	50	85	415



Рис. 8.40



Рис. 8.41

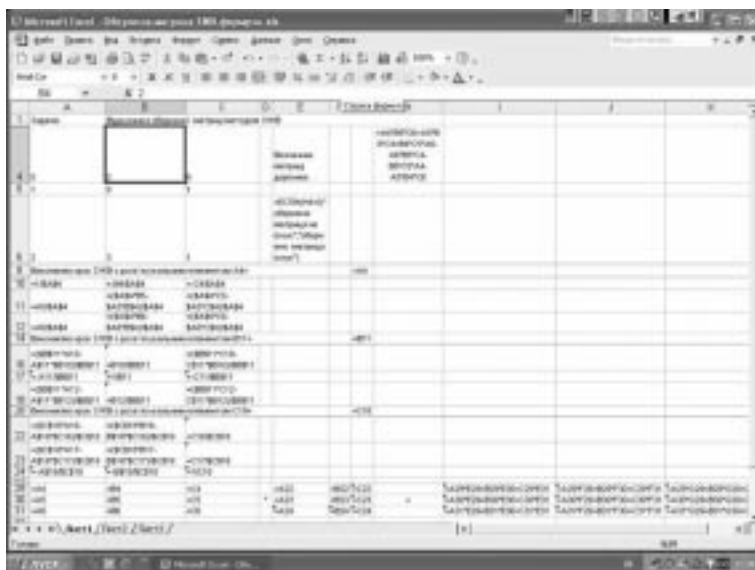


Рис. 8.42

Знайти обсяг валового випуску для кожної галузі, якщо кінцевий продукт першої галузі має збільшитися вдвічі, а другої — на 30 %.

Нехай x_{ij} — обсяг продукції i -ї галузі, використаної j -ю галуззю в процесі виробництва ($i, j = 1, 2$); x_i — загальний (валовий) обсяг продукції i -ї галузі ($i = 1, 2$). Тоді $x_{11} = 120$; $x_{12} = 180$; $x_{21} = 280$; $x_{22} = 50$; $x_1 = 540$; $x_2 = 415$. За умовою задачі кінцевий продукт першої галузі має збільшитися вдвічі, тобто з 240 до 480, а другої — на 30 %, тобто з 85 до 110,5. Вектор кінцевого продукту —

$$Y = \begin{pmatrix} 480 \\ 110,5 \end{pmatrix}.$$

Коефіцієнти прямих витрат a_{ij} ($i, j = 1, 2$) можна обчислити за формулою

$$a_{ij} = \frac{x_{ij}}{x_j} \quad (i, j = 1, 2):$$

$$a_{11} = \frac{120}{540} \approx 0,22; \quad a_{12} = \frac{180}{415} \approx 0,43;$$

$$a_{21} = \frac{280}{540} \approx 0,52; \quad a_{22} = \frac{50}{415} \approx 0,12.$$

Матриця прямих витрат $A = \begin{pmatrix} 0,22 & 0,43 \\ 0,52 & 0,12 \end{pmatrix}$ має невід'ємні елементи й задовольняє критерій продуктивності:

$$\max\{0,22 + 0,52; 0,43 + 0,12\} = \max\{0,74; 0,55\} = 0,74 < 1.$$

Тому для кожного вектора кінцевого продукту Y можна знайти потрібний обсяг валового випуску X за формулою

$$X = (E - A)^{-1} Y.$$

Визначити матрицю $E - A$ легко:

$$E - A = \begin{pmatrix} 0,78 & -0,43 \\ -0,52 & 0,88 \end{pmatrix}.$$

Обчислимо матрицю повних витрат $S = (E - A)^{-1}$ за допомогою стандартних функцій Excel. Розглянемо матрицю $E - A$ (діапазон А1 : В2). Перевіримо, чи існує для неї обернена матриця. Для цього виконаємо такі дії:

- 1) перемістимо курсор у клітинку D1;
- 2) із головного меню виберемо команду “Вставка” — “Функція”;
- 3) із запропонованих категорій виберемо “Математичні”, а із запропонованих функцій — МОПРЕД;
- 4) на запит системи введемо за допомогою “миші” діапазон А1 : В2 та оберемо “ОК” для закінчення введення (рис. 8.43).

У клітинку D1 заноситься число, що дорівнює визначнику матриці $E - A$. Оскільки отримане значення визначника (0,4628) не дорівнює нулю, обернена матриця $S = (E - A)^{-1}$ існує. Нагадаємо, як обчислюється матриця S :

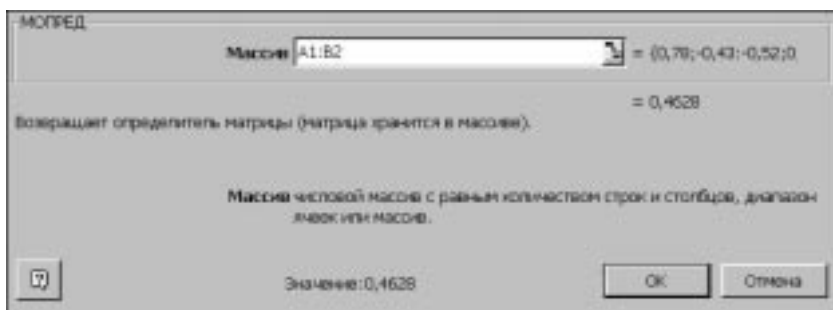
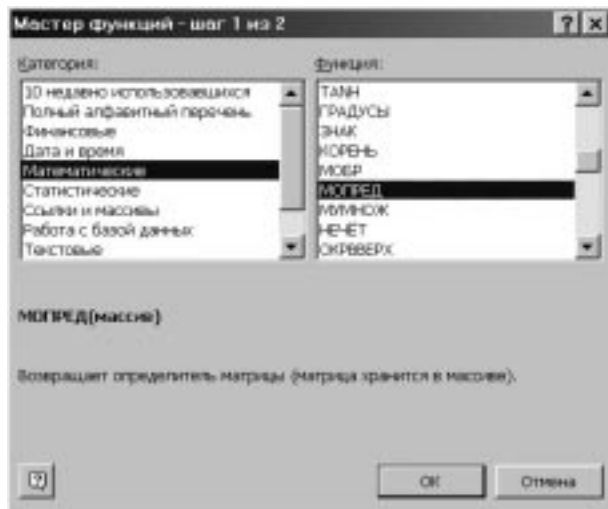


Рис. 8.43

1) виділимо за допомогою клавіш Shift і стрілок діапазон A4 : B5 для розміщення матриці $(E - A)^{-1}$;

2) виберемо з головного меню команду “Вставка” — “Функція”; із запропонованого набору категорій виберемо “Математичні”, із набору функцій — МОБР;

3) у відповідь на запит уведемо діапазон A1 : B2 (за допомогою “миші”);

4) закінчимо виконання команди комбінацією клавіш Shift + Ctrl + Enter (обов’язково).

У діапазоні A4 : B5 записується матриця $S = (E - A)^{-1}$ (рис. 8.44).

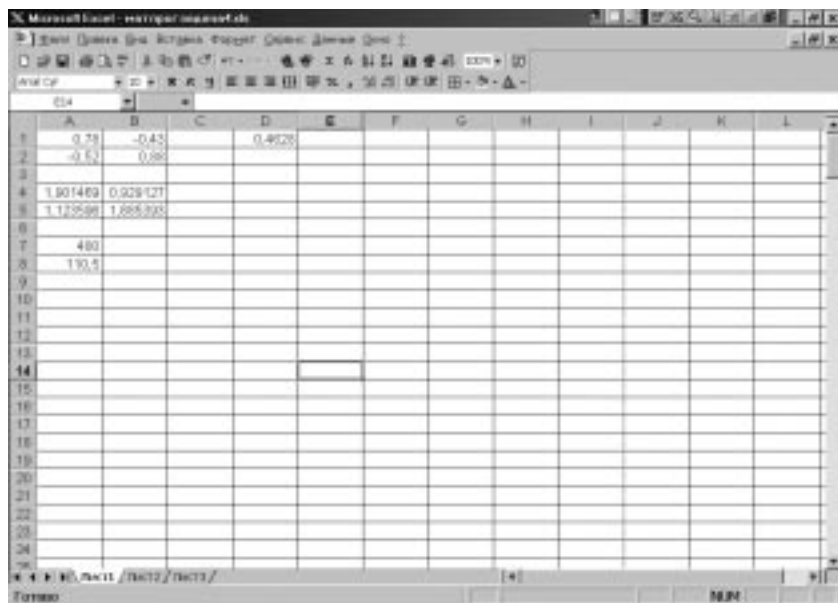
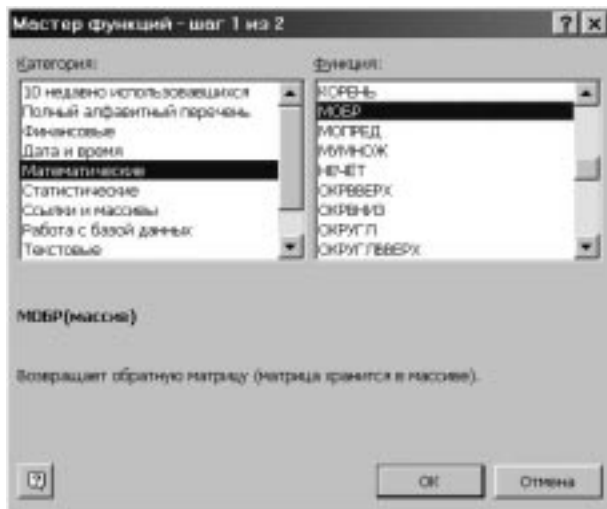


Рис. 8.44

З використанням формули $X = (E - A)^{-1} Y$ можна знайти вектор X валового випуску. Для цього в діапазон A7 : A8 уведемо матрицю Y і помножимо матриці $(E - A)^{-1}$ та Y за допомогою функції множення матриць МУМНОЖ.

Очевидно, що результат множення матриць розмірністю 2×2 та 2×1 — матриця-стовпець розмірністю 2×1 , тому перед зверненням до функції МУМНОЖ потрібно виділити діапазон клітинок, у якому передбачається розмістити результат множення (у цьому разі A10 : A11).

На запит системи щодо множників потрібно ввести діапазони A4 : B5 і A7 : A8, у яких розміщуються відповідно матриці $(E - A)^{-1}$ та Y .

Отриманий у діапазоні A10 : A11 вектор $X = \begin{pmatrix} 1015,374 \\ 725,5618 \end{pmatrix}$ — розв’язок

задачі (рис. 8.45).

Отже, обсяг валового випуску в першій галузі потрібно збільшити з 540 до 1015,4 умовних грошових одиниць, а у другій галузі — до 725,6 умовних одиниць.

Ефективним інструментом розв’язання багатьох оптимізаційних задач є програма Excel “Поиск решения” з меню “Сервис”. Вона дає змогу за заданим значенням результату (мінімальним чи максимальним) знаходити множину значень змінних, які задовольняють деякі обмеження. У програмі реалізовано універсальний метод розв’язання задач лінійного програмування — симплексний метод. Застосування цієї програми дає змогу користувачам, які недостатньо володіють методом, легко розв’язувати оптимізаційні задачі, зокрема ЗЛП, ЗЦП, транспортні задачі тощо.

Розглянемо приклад застосування програми “Поиск решения” для розв’язання наступної цілочислової ЗЛП.

Знайти максимум цільової функції

$$F(x) = 15x_1 + 22x_2 + 19x_3$$

за умов

$$2x_1 + 3x_2 + 2x_3 \leq 360,$$

$$x_1 + 2x_2 + 3x_3 \leq 240,$$

$$x_1 + x_2 + 2x_3 \leq 180,$$

$x_1, x_2, x_3 \geq 0$, — цілі числа.

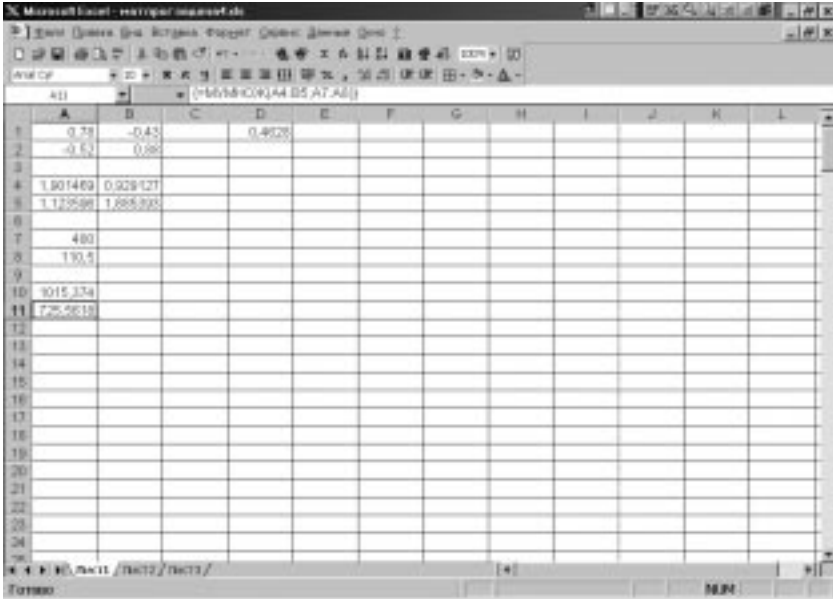
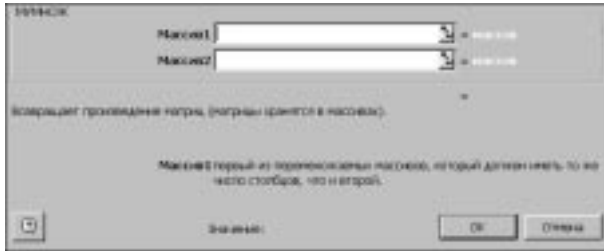


Рис. 8.45

Початкові дані будемо вводити у вигляді таблиць.

У комірках A2 : A4 запишемо позначення змінних x_1 , x_2 , x_3 .

У комірках B2 : B4 запишемо довільні невід’ємні значення змінних x_1 , x_2 , x_3 (наприклад, нульові). Ці значення будуть змінюватися впродовж обчислень, а після їх закінчення, там буде зберігатись розв’язок задачі.

Матрицю коефіцієнтів системи обмежень запишемо у комірках C2 : E2 у транспонованому вигляді, тобто

$$A^T = \begin{pmatrix} 2 & 1 & 1 \\ 3 & 2 & 1 \\ 2 & 3 & 2 \end{pmatrix}.$$

У діапазоні C5 : E5 запишемо вектор правих частин системи обмежень задачі, тобто значення (360, 240, 180), а коефіцієнти цільової функції (15, 22, 19) розташуємо у діапазоні H2 : H4.

У комірці A8 запишемо значення цільової функції задачі, що визначається за допомогою вбудованої математичної функції СУММПРОИЗВ(массив1, массив2). Перший масив — це значення коефіцієнтів цільової функції (діапазон H2 : H4), а другий — значення змінних (діапазон B2 : B4). Отже, у комірці A8 маємо формулу “=СУММПРОИЗВ(H2 : H4; B2 : B4)”.

У діапазоні C8 : E8 вводимо формули для обчислення обмежень задачі, а саме у комірці C8 набираємо “=СУММПРОИЗВ(C2 : C4; \$B\$2 : \$B\$4)” і продовжуємо її у комірці D8, E8.

Потім у меню “Сервис” вибираємо “Поиск решения” і відповідаємо на запити програми, вказуючи, “мишею” потрібні комірки чи їх діапазони:

“Установить целевую ячейку” — вказуємо \$A\$8;

“Равной” — вибираємо “максимальному значению”;

“Изменяя ячейки” — \$B\$2:\$B\$4;

“Ограничения” — \$B\$2:\$B\$4 = целое

\$B\$2:\$B\$4 >=0

\$C\$8 <= \$C\$5

\$D\$8 <= \$D\$5

\$E\$8 <= \$E\$5

Кнопки “Добавить”, “Изменить”, “Удалить” використовуються для коригування обмежень задачі. Кнопка “Выполнить” розпочинає процедуру пошуку розв’язку.

Після виконання розрахунків з’являється діалогове вікно “РЕЗУЛЬТАТЫ ПОИСКА РЕШЕНИЯ”. Якщо знайдено розв’язок, то маємо повідомлення “Решение найдено. Все ограничения и условия оптимальности выполнены”.

Результати розрахунків будуть збережені, якщо вибрати варіант “Сохранить найденное решение” і натиснути кнопку “ОК”. Результати будуть знищені, якщо натиснути кнопку “Отмена”.

Щоб отримати звіт про роботу програми “Поиск решения” і результати її розрахунків, потрібно у запиті “Тип отчета” вибрати “результаты”. У звіті виводиться інформація про значення цільової функції і її адресу, про початкові і кінцеві значення керованих змінних, про обмеження задачі. Якщо в коментарях обмежень вказано статус “связанный”, то це означає, що обмеження виконується як рівність, тобто відповідна змінна набула свого граничного значення, яке далі змінюватися не може.

Якщо умови задачі потрібно змінити, то у діалоговому вікні “ПОИСК РЕШЕНИЯ” у запиті “Ограничения” їх коригують за допомогою кнопок “Добавить”, “Изменить” або “Удалить”. Після команди “Выполнить” отримують новий розв’язок ЗЛП.

Якщо у результаті виконання процедури “Поиск решения” розв’язок не знайдений, то потрібно перевірити умови задачі. В окремих випадках їх можна скоригувати, змінивши параметри процедури (кнопка “Параметры” у діалоговому вікні “ПОИСК РЕШЕНИЯ”).

СПИСОК ВИКОРИСТАНОЇ ТА РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ

1. *Акулич И. Л.* Математическое программирование в примерах и задачах. — М.: Высш. шк., 1986.
2. *Вагнер Г.* Основы исследования операций: В 3 т. — М.: Мир, 1973.
3. *Вентцель Е. С.* Исследование операций. — М.: Сов. радио, 1972.
4. *Гетманцев В. Д.* Лінійна алгебра і лінійне програмування. — К.: Либідь, 2001.
5. *Жалдак М. І.* Комп'ютер на уроках математики: Посіб. для вчителів. — К.: Техніка, 1997.
6. *Зайченко Ю. П.* Дослідження операцій. — К.: ВІПОЛ, 2000.
7. *Зайченко Ю. П.* Исследование операций: Учеб. для вузов. — К.: Выща шк., 1975.
8. *Карманов В. Г.* Математическое программирование. — М.: Наука, 1975.
9. *Конюховский П. В.* Математические методы исследования операций в экономике. — СПб.: Питер, 2000.
10. *Кудрявцев Е. М.* Исследование операций в задачах, алгоритмах, программах. — М.: Радио и связь, 1984.
11. *Линейное и нелинейное программирование / И. Н. Ляшенко и др.* — К.: Выща шк., 1975.
12. *Мазаракі А. А., Толбатов Ю. А.* Математичне програмування в Excel: Навч. посіб. — К.: Четверта хвиля, 1998.
13. *Математические методы исследования операций: Учеб. пособие для вузов / Ю. М. Ермольев, И. И. Ляшко, В. С. Михалевич, В. И. Тюптя.* — К.: Выща шк., 1979.
14. *Математическое программирование / Ю. Н. Кузнецов и др.* — М.: Высш. шк., 1980.
15. *Попов Ю. Д.* Линейное и нелинейное программирование: Учеб. пособие. — К.: Изд-во КГУ, 1988.
16. *Таха Хэмди А.* Введение в исследование операций: Пер. с англ. — 6-е изд. — М.: Изд. дом "Вильямс", 2001.

ЗМІСТ

<i>Вступ</i>	3
1. Лінійне програмування	9
2. Транспортна задача	53
3. Цілочислові та дискретні задачі лінійного програмування	68
4. Нелінійне програмування	84
5. Динамічне програмування	109
6. Задачі та методи стохастичного програмування	117
7. Матричні ігри	123
8. Розв'язування деяких задач математичного програмування з використанням інформаційних технологій	135
<i>Список використаної та рекомендованої літератури</i>	181

The manual contains a necessary minimum of information from the theory of optimization. The basic tasks of optimizations, which arise up during making organizational decisions are examined.

The traditional sections of the mathematical programming are given: linear and nonlinear programming, transport problem and tasks of integer linear programming. The design of dynamics of economic processes is expounded on the examples of tasks of the dynamic programming. Optimization of decisions in the conditions of risk and vagueness is examined on the examples of the stochastic programming tasks and matrix games. Some possibilities of modern programmatic facilities using are exposed for the decision of the mathematical programming tasks.

For the students of economic specialties, who has a certain level of higher mathematics, algebra and theory of chances knowledge.

Навчальне видання

Жильцов Олексій Борисович

Кулян Віктор Романович

Юнькова Олена Олександрівна

МАТЕМАТИЧНЕ ПРОГРАМУВАННЯ

(з елементами інформаційних технологій)

Навчальний посібник

Educational edition

Zhil'tsov, Oleksij B.

Kulian, Victor R.

Un'kova, Olena O.

MATHEMATICAL PROGRAMMING

(with the informational technologies elements)

Educational manual

Відповідальний редактор *С. Г. Рогузько*

Редактор *С. Г. Атадєва*

Комп'ютерне верстання *Н. М. Музиченко*

Оформлення обкладинки *Д. В. Кругленко*

Підп. до друку 16.01.06. Формат 60×84¹/₁₆. Папір офсетний. Друк офсетний.

Ум. друк. арк. 10,69. Обл.-вид. арк. 9,45. Тираж 5000 пр. Зам. № 32

Міжрегіональна Академія управління персоналом (МАУП)

03039 Київ-39, вул. Фрометівська, 2, МАУП

*Свідоцтво про внесення до Державного реєстру
суб'єктів видавничої справи ДК № 8 від 23.02.2000*

Поліграфічний центр УТОГ

03038 Київ-38, вул. Нововокзальна, 8

Свідоцтво КІ № 35 від 02.08.2000