МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
"КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО"

# Дослідження операцій та методи оптимізації в біології та медицині

## Навчальний посібник

*Рекомендовано Вченою радою КПІ ім. Ігоря Сікорського*
як навчальний посібник для здобувачів ступеню бакалавра
за освітньою програмою "Комп'ютерні технології в біології та медицині"
спеціальності "Комп'ютерні науки"

Електронне мережне навчальне видання

Київ
КПІ ім. Ігоря Сікорського
2023

Укладачі: Городецька О. К., к.т.н., доцент
Зеленський К. Х., д.т.н., доцент
Настенко Є. А., д.б.н., к.т.н., професор
Павлов В. А., к.т.н., доцент

**Рецензенти:**

*Синєглазов В. М., д.т.н., професор, завідувач кафедри авіаційних комп'ютерно-інтегрованих комплексів*
Національного авіаційного університету

*Сільвестров А. М., д.т.н., професор*
Національний технічний університет України
"Київський політехнічний інститут імені Ігоря Сікорського"

**Відповідальний редактор**
*Білошицька О. К., к.т.н., доцент*
Національний технічний університет України
"Київський політехнічний інститут імені Ігоря Сікорського"

*Гриф надано Методичною радою КПІ ім. Ігоря Сікорського (протокол № 5 від 23.02.2023 р.) за поданням*
*Вченої ради факультету біомедичної інженерії (протокол № 6 від 30. 01. 2023 р.)*

У навчальному посібнику "Дослідження операцій та методи оптимізації в біології та медицині" викладено загальні питання задач лінійного та нелінійного програмування. Доклано викладаютья питання оптимізації задач лінійного програмування: транспортна задача, симплекс-метод, дискретне і квадратичне програмування. Викладено вирішення задач оптимізації нелінійних функцій без обмежень та з обмеженнями. Розглянуто застосування методів лінійного та нелінійного програмування стосовно проблем у біолоігї та медицині. Викладання теоретичного матеріалу ілюструється численними прикладами їхнього застосування.

Реєстр. № НП 22/23-510. Обсяг 12 авт. арк.
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
проспект Перемоги, 37, м. Київ, 03056
https://kpi.ua
Свідоцтво про внесення до Державного реєстру видавців, виготовлювачів і розповсюджувачів видавничої продукції ДК № 5354 від 25.05.2017 р.

© О.К. Городецька, К.Х. Зеленський, Є.А. Настенко, В.А. Павлов
© КПІ ім. Ігоря Сікорського, 2023

# Зміст


| Розділ | Назва розділу | Стор. |
|---|---|---|
| | Передмова | 6 |
| 1 | **Загальні питання дослідження операцій** | 7 |
| | 1.1. Вступна частина | 7 |
| | 1.2. Приклади математичних моделей | 8 |
| | 1.2.1. Визначення найкращого складу сумішей | 8 |
| | 1.2.2. Задача про оптимальний план випуску продукції | 9 |
| | 1.2.3. Оптимізація міжгалузевих потоків | 9 |
| | 1.2.4. Транспортна задача | 10 |
| | 1.2.5. Параметрична транспортна задача | 11 |
| | 1.2.6. Нелінійна задача розташування | 11 |
| | 1.2.7. Цілочислова задача розташування | 12 |
| | 1.3. Постановка задачі математичного програмування | 12 |
| | 1.3.1. Зв'язок задачі нелінійного програмування з реальними процесами | 13 |
| | 1.3.2. Випуклість | 14 |
| | 1.3.3. Теорема Куна-Такера | 14 |
| | 1.4. Загальна постановка задач прийняття рішень | 16 |
| 2 | **Лінійне програмування** | 16 |
| | 2.1. Задача лінійного програмування | 19 |
| | 2.2. Симплекс-метод | 23 |
| | 2.2.1. Визначення базисного розв’язання | 23 |
| | 2.2.2. Пошук опорного розв’язання | 25 |
| | 2.2.3. Симплекс-таблиця | 26 |
| | 2.2.4. Приклади | 28 |
| | 2.3. Канонічна форма задачі ЛП | 31 |
| | 2.4. Транспортні задачі і методи їхнього розв’язання | 34 |
| | 2.4.1. Метод потенціалів | 38 |
| | 2.5. Цілочислове лінійне програмування | 42 |
| | 2.5.1. Постановка задачі | 44 |
| | 2.5.2. Метод Гоморі | 46 |
| | 2.5.3. Дискретне програмування | 50 |
| | 2.6. Комбінаторні методи у дискретному програмуванні | 52 |
| | 2.6.1. Метод гілок та меж | 54 |
| | 2.7. Апроксимуюче лінійне програмування | 56 |
| | 2.8. Квадратичне програмування | 61 |
| | 2.8.1. Метод Вульфа | 62 |

| | | |
|---|---|---|
| | 2.9. Двоїстість у задачах ЛП | 63 |
| | 2.10. Завдання для самостійної роботи | 64 |
| **3** | **Методи оптимізації нелінійних функцій без обмежень** | 67 |
| | 3.1.Постановка задачі | 67 |
| | 3.2. Методи одновимірного пошуку «золотого перетину» | 68 |
| | 3.2.1. Метод «золотого перетину» | 69 |
| | 3.2.2. Метод Фібоначчі | 69 |
| | 3.2.3. Метод сполучених напрямків Пауела | 70 |
| | 3.3. Градієнтні методи | 73 |
| | 3.3.1. Метод найскорішого спуску | 74 |
| | 3.4. Метод сполучених градієнтів | 76 |
| | 3.5. Метод Флетчера—Рівса | 78 |
| | 3.5.1. Реалізація методу Флетчера-Рівса | 79 |
| | 3.6. Метод Ньютона | 79 |
| | 3.6.1. Реалізація методу Ньютона | 82 |
| | 3.7. Метод Девідона—флетчера—Пауела | 83 |
| | 3.7.1. Реалізація методу ДФП | 85 |
| | 3.8. Контрольні запитання та задачі | 87 |
| **4** | **Задачі із обмеженнями** | 90 |
| | 4.1. Методи спуску | 90 |
| | 4.1.1. Метод проекції градієнту | 90 |
| | 4.1.2. Метод допустимих напрямків | 94 |
| | 4.2. Методи штрафних функцій | 96 |
| | 4.2.1. Метод узагальнених мноників Лагранжа | 97 |
| | 4.2.2. Метод послідовної безумовної мінімізації | 99 |
| | 4.2.3. Реалізація МПБМ | 102 |
| **5** | **Динамічне програмування** | 108 |
| | 5.1. Дискретні керуючі системи | 111 |
| | 5.2. Загальна схема динамічного програмування | 112 |
| | 5.2.1. Числова реалізація | 114 |
| | 5.3. Приклади | 115 |
| | 5.4. Метод ДП для неперервних систем | 122 |
| | 5.4.1. Автономна система | 122 |
| | 5.4.2. Неавтономна система | 124 |
| | 5.5. Контрольні запитання і задачі | 125 |
| **6** | **Застосування методів оптимізації в біології та медицині** | 126 |
| | 6.1. Передумови для застосування методів оптимізації | 126 |
| | 6.1.1. Спільне та відмінності в задачах ЛП та МНК | 127 |

| | | |
|---|---|---|
| | 6.2. Підходи до вирішення несумісних оптимізаційних задач | 128 |
| | 6.2.1. Метод безумовної оптимізації за несумісної системи обмежень | 128 |
| | 6.2.2. Застосування задач ЛП при несумісній системі обмежень | 128 |
| | 6.2.3. Завдання моделювання з мінімізацією критерію модулю нев’язки | 129 |
| | 6.2.4. Завдання моделювання з одностороннім критерієм та мінімаксом | 130 |
| | 6.3. Розрахунок оптимальної виробничої програми | 131 |
| | 6.4. Розрахунок оптимальних лікувальних стратегій | 134 |
| | 6.4.1. Обгрунтування завдання | 134 |
| | 6.4.2. Постановка завдання | 134 |
| | 6.4.3. Шляхи і труднощі при вирішенні проблеми | 135 |
| | 6.4.4. Розрахунок у класі лінійних стратегій | 137 |
| | 6.5. Контрольні питання | 138 |
| | | |

# Передмова

Математичне програмування належить до числа дисциплін, що найчастастіше застосовуються у прикладній математиці. Саме тому цей курс включено до більшості навчальних планів вищих навчальних закладів. Незважаючи на те, що математичне програмування, як окрема дисципліна, виникло для розв'язання задач економічного характеру, ця дисципліна стала базовою при вирішенні різноманітних задач багатьох технічних галузей, в яких у тій чи іншій формі виникає необхідність у вирішенні задач оптимізації тієї чи іншої функції мети за наявності чи відсутності обмежень.

Незважаючи на доволі великий обсяг наукової та навчальної літератури, яка присвячена окремим розділам математичного програмування (найбільш поширеними та змістовними є джерела, в яких викладаються методи вирішення задач лінійного програмування), все ж таки важко порекомендувати посібник, у якому були б розглянуті у досить повному обсязі, як задачі лінійного, так і нелінійного програмування.

Крім того, враховуючи сучасний рівень обчислювальної техніки, доцільно викладати методи та алгоритми, що їх реалізують, орієнтуючись на програмну реалізацію цих алгоритмів тією чи іншою алгоритмічною мовою. У навчальному посібнику, як таку мову, обрано алгоритмічну мову С тому, що вона є проблемно-орієнтованою та оптимальною з точки зору її реалізації сучасними компіляторами у машинний код.

# Розділ 1. Загальні питання дослідження операцій

## 1.1. Вступна частина

З метою вивчення та дослідження складних об'єктів, явищ часто застосовується моделювання. Найпростішим засобом моделювання є метод подібності шляхом відтворювання явища в експериментальних умовах зі зміненим масштабом. Модель - це спеціально побудований об'єкт, на якому відтворюються цілком визначені характеристики об'єкту з метою його вивчення, а моделювання - цілком визначене відображення характеристик об'єкту, що вивчається, з метою його дослідження. Моделювання є важливим інструментом наукової абстракції, що надає можливість виділити, відокремити та аналізувати суттєві для даного дослідження об'єкту характеристики - властивості, взаємні зв'язки, структурні та функціональні параметри.

Ступінь відповідності моделі та об'єкту моделювання може бути різною. Наприклад, функціональні моделі відображують лише поведінку, функцію об'єкта. Вони широко використовуються для моделювання "чорного ящика" у кібернетиці. Під "чорним ящиком" мається на увазі система, внутрішні складові якої невідомі спостерігачу й не мають для нього значення, але ж він може досліджувати "входи" та "виходи" системи. Моделі можуть бути реалізовані як за допомогою деяких фізичних об'єктів (фізична модель, наприклад, тренажери, моделі діючих об'єктів), так і за допомогою абстрактних об'єктів. Такою абстрактною моделлю є математична модель, що описується деяким набором математичних виразів або рівнянь тієї чи іншої природи.

Математичне моделювання є найдовершенішим й найефективнішим методом моделювання. Цілком природно, що результати досліджень такої моделі можуть мати практичний сенс, якщо модель буде досить адекватною явищу, що розглядається, тобто досить повно відображати реальну ситуацію.

У зв'язку з дедалі ширшим залученням математичного апарату для дослідження й розв'язання економічних задач виникла необхідність у розробці методів, що навмисне пристосовані до задач економічного аналізу. Саме завдяки цьому виникла низка нових математичних дисциплін, таких, як лінійне програмування, динамічне програмування, теорія ігор, теорія графів тощо.

Предметом дослідження цих прикладних дисциплін є математичні моделі, які пов'язані у своїй більшості із визначеними економічними процесами, що описують динаміку підприємств, промислових об'єднань тощо.

Особливу роль відіграють питання пошуку найкращого в деякому сенсі, або оптимального плану. Математичні методи пошуку найкращого, оптимального розв'язання об'єднані під назвою "математичне програмування". Цей термін виник під впливом праць Роберта Дорфмана у 1950 році стосовно задач лінійного програмування. У подальшому під задачами математичного програмування стали розуміти задачі, в яких відшукується мінімум або максимум деякої функції за наявності обмежень на незалежні змінні, від яких залежить функція й для якої відшукується екстремум.

Функція, для якої відшукується екстремум (мінімум чи максимум), має назву функції мети, а обмеження на незалежні змінні, що мають місце у математичній моделі, являють собою систему відношень, яка звужує *область допустимих значень* (ОДЗ) так званих керованих

змінних, тобто величин, що вимірюються, й значення яких підлягають оптимізації.

Функція мети, що виражається через керовані змінні та обмеження, й складає математичну модель задачі оптимізації. Будь-який вибір значень змінних, що задовольняють обмеженням, визначає *допустимий* план, а той з них, на якому досягається максимум або мінімум, визначає *оптимальний план*.

## 1.2. Приклади математичних моделей

### 1.2.1. Визначення найкращого складу суміші

Однією з перших задач, розв'язання яких одержано Дж. Данцігом методами математичного (лінійного) програмування, була задача про вибір дієти.

Нехай нам відомий склад харчових речовин, що необхідні для харчування тварин у різноманітних кормах. Відома також вартість одиниці кожного виду корму. Необхідно обрати раціон - набір та кількість корму - таким чином, щоб кожна речовина містилась у ньому в необхідній кількості, що забезпечує харчування тварини й, окрім цього, щоб сумарні витрати на цей раціон були мінімальними. Позначимо: $m$ - кількість різних необхідних харчових речовин, $n$ — кількість видів кормів,

$a_{ik}$ -- кількість одиниць $i$-ї харчової речовини, що міститься в одиниці $k$-го виду кормів, $b_i$ - мінімальна потреба за добу в і-й харчовій речовині, $c_k$ - вартість одиниці $k$-го виду корму, $x_k$ — кількість одиниць $k$-го виду корму, що використовується у раціоні, й підлягає визначенню. Оптимальним планом у цій задачі є така упорядкованість чисел $x_1, x_2, \ldots, x_n$, що має задовольняти таким обмеженням:

1. $x_k \geq 0$, $k = \overline{1,n}$;

$$2. \sum_{i=1}^{n} a_{ij} x_i \geq d_i, \quad i = \overline{1,n}.$$

(загальна кількість харчової речовини у даному раціоні не нижче заданої).

3. Мінімізувати сумарні витрати на складання оптимального раціону:

$$\min \sum_{i=1}^{n} c_k x_k.$$

Аналогічним є приклад про склад шихти. Відомо, що для отримання легірованої сталі необхідно використовувати шихту визначеного хімічного складу. Більшість інгредієнтів шихти коштовні. У той же час склад шихти містить малокоштовні матеріали - чавун, лом, відходи із визначеним вмістом присадок. Виникає задача про вибір такої шихти, до складу якої входили б у заданій кількості необхідні хімічні речовини, й її коштовність була б мінімальною. Математична модель цієї задачі збігається із моделлю задачі про вибір дієти.

Бензини різних сортів отримують змішуванням нафтопродуктів, що мають різні технічні характеристики. Визначені показники якості бензину (октанове число, ступінь очистки тощо) мають дотримуватись досить точно, оскільки вони відіграють важливу роль у використанні бензину. Від того, які нафтопродукти при цьому змішуються, залежить рентабельність виробництва. Необхідно побудувати такий план змішування нафтопродуктів, який має забезпечити максимальну рентабельність виробництва - з одного боку, та дозволяв би отримувати бензини різних сортів у необхідних пропорціях - з іншого.

### 1.2.2. Задача про оптимальний план випуску продукції

Нехай номенклатура продукції, що виробляється підприємством, складається із п найменувань. Позначимо через $a_{ij}$ витрати $i$-го виду ресурсів $(i=\overline{1,m})$ на виробництво одиниці продукції $j$-го виду $(j=\overline{1,n})$, через $b_i$ -- повні обсяги наявних ресурсів, $c_j$ -- прибуток, що одержується підприємством при виготовленні та реалізації одиниці $j$-го виду продукту, $a_j$, $A_j$ -- відповідно заздалегідь задані нижню та верхню межі обсягів випуску $j$-то виду продукції.

Необхідно скласти такий план випуску продукції, який було би технологічно здійснено за наявними ресурсами всіх видів, й який задовольняв би обмеженням, що задані на випуск кожного виду продукції, та в той же час забезпечував найбільший загальний прибуток підприємству.

Математична модель полягає у такому:

знайти такий план випуску продукції $x=[x_1,x_2,\ldots,x_n]$, щоб задовольнялись нерівності:

$$1.\ \sum_{j=1}^{n} a_{ij}x_j, \quad (i=\overline{1,m})$$

(технологічні обмеження):

$$2.\ a_j \le x_j \le A_j, \quad (j=\overline{1,n})$$

(обмеження на обсяги окремих видів продукції, що виготовляється)

й при цьому досягався б

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

(загальний прибуток від виробництва та реалізації продукції).

### 1.2.3. Оптимізація міжгалузевих потоків

Нехай є $n$ галузей господарства, кожна з яких виробляє тільки один специфічний вид продукції, причому кожний вид продукції, що вироблено, використовується у виробництві в усіх $n$ галузях. Нехай $x_i$ -- обсяг виробництва в $i$-й галузі, $y_i$ -- обсяг продукту для використання у невиробничій сфері, $a_{ij}$ -- коефіцієнти прямих затрат продукції $j$-го виду на виробництво у $i$-й галузі одиниці продукції $i$-го виду, $N_i$ -- максимально можливий обсяг виробництва у $i$-й галузі, $d_i$ -- кількість продукції $i$-го виду, що потребує невиробнича сфера, $c_i$ -- вартість одиниці продукції $i$-го виду.

Необхідно знайти такі можливі у заданих умовах обсяги виробництва та такий план випуску кінцевої продукції $y_i$, $(i=\overline{1,m})$, за якого максимізується загальна вартість кінцевого продукту.

Математична модель цієї задачі може бути записана у такому вигляді. Знайти такі

вектори $x = [x_1, x_2, \ldots, x_n]$ і $y = [y_1, y_2, \ldots, y_n]$, щоб досягався

$$\max \sum_{j=1}^{n} c_j y_j$$

(загальна вартість кінцевого продукту) за умови виконання обмежень:

1. $0 \le x_i \le N_i$ (обмеження на обсяги виробництва);
2. $y_i \ge d_i$, $i = \overline{1,n}$ (обмеження на випуск кінцевого продукту):
3. $$x_i \ge \sum_{j=1}^{n} a_{ij} x_i + y_i, \quad (i = \overline{1,n})$$

(технологічні обмеження на випуск продукції).

### 1.2.4. Транспортна задача

У найпростішому варіанті ця задача виникає, коли мова йде про раціональне перевезення деякого однорідного продукту від виробника до споживача. Вважається, що споживачам байдуже, звідки, із яких пунктів виробництва буде надходити продукт, аби він надходив у необхідній кількості. Але ж від того, наскільки раціональним буде прикріплення пунктів споживання до пунктів виробництва, суттєво залежить обсяг транспортної роботи. Задача формулюється наступним чином:

Нехай є $m$ пунктів виробництва з обсягами виробництва за одиницю часу (місяць, квартал тощо), що дорівнюють $a_1, a_2, \ldots, a_m$, та $n$ пунктів споживання з обсягами споживання $\theta_1, \theta_2, \ldots, \theta_n$. Природно вважати, що $\sum_{i=1}^{m} a_i \ge \sum_{j=1}^{n} \theta_j$, тобто споживання не перевищує можливості виробництва.

Відомі величини $c_{ij}$ -- витрати на перевезення одиниці продукту з $i$-го пункту виробництва у $j$-й пункт споживання. Необхідно скласти такий план перевезення? щоб задовольнити потреби в усіх $n$ пунктрах споживання і при цьому сумарні витрати на перевезення були б мінімальними.

Позначимо через $x_{ij}$ кількість продукту, що перевозиться з $i$-го пункту виробництва у $j$-й пункт споживання. Тоді маємо таку математичну модель.

Знайти такі значення величин , щоб досягався

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

(сумарні витрати на транспортування) за умов:

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

(у кожний пункт споживання завозиться потрібна кількість продукту):

$$2. \sum_{j=1}^{m} x_{ij} \le a_i, \quad i = \overline{1,m}$$

(з кожного пункту виробництва вивозиться кількість продукту, що не перевищує кількості виробленого продукту):

$$3.\ x_{ij} \ge 0, \quad j = \overline{1,n};\ i = \overline{1,m}$$

(обсяг продукту, що перевозиться, не може бути від'ємним).

### 1.2.5. Параметрична транспортна задача

Нехай потужність постачальників дорівнює $a_i$, попит споживачів -- $]theta_j$ і витрати на перевезення одиниці продукції $c_{ij}$ лінійно змінюються залежно від параметру $t$ (приміром, часу):

$$a_i = \alpha_i + \alpha_i' t;\ i = \overline{1,m};$$
$$c_j = \beta_j + \beta_j' t;\ j = \overline{1,n};$$
$$c_{ij} = \gamma_{ij} + \gamma_{ij}' t;\ i = \overline{1,m};\ j = \overline{1,n}.$$

Необхідно мінімізувати транспортні витрати

$$\min\left\{\sum_{i=1}^{m}\sum_{j=1}^{n}(\gamma_{ij} + \gamma_{ij}' t)x_{ij}\right\}$$

із обмеженнями:

$$1.\ \sum_{j=1}^{m} x_{ij} \le \alpha_i + \alpha_i' t;\ i = \overline{1,m};$$
$$2.\ \sum_{i=1}^{n} x_{ij} \le \beta_j + \beta_j' t;\ j = \overline{1,n};$$
$$3.\ x_{ij} \ge 0;\ i = \overline{1,m};\ j = \overline{1,n}.$$

### 1.2.6. Нелінійна задача розташування

Нехай $f_i(x_i)$ -- функція, що являє собою витрати на виробництво продукту в обсязі на г-му підприємстві. Виробничі потужності не обмежуються. За таких припущень одержуємо таку задачу.

Знайти такі значення змінних , щоб досягався

$$\min\left\{\sum_{i=1}^{m} f_i(x_i) + \sum_{i=1}^{m}\sum_{j=1}^{n}(\gamma_{ij} + \gamma_{ij}' t)x_{ij}\right\}$$

(мінімум сумарних витрат на виробництво та транспортування, $C$ -- матриця транспортних витрат) за умов:

$$1.\ x_{ij} \ge 0, i = \overline{1,m};\ j = \overline{1,n}$$

(кількості продукту, що вироблено, є невід'ємними);

$$2.\ x_i = \sum_{j=1}^{n} x_{ij}$$

(кількість продукту, що вироблено, сповна вивозиться й доставляється споживачам);

3. $\sum_{i=1}^{m} x_{ij} \geq \theta_j$

(кожний пункт споживача одержує не менше, ніж заданий у цьому пункті обсяг споживання).

### 1.2.7. Цілочислова задача розташування

Нехай є $m$ пунктів, в яких можуть бути розташовані підприємства, що виробляють деякий продукт. Передбачається, що підприємства створюються за типовими проектами й їхні виробничі потужності $x_i$ можуть набувати лише обмежену множину значень. Приміром, підприємство може бути оснащене однією, двома або трьома конвеєрними лініями, мати два, чотири або шість однотипних агрегатів (печі, котли тощо). Таким чином, у даному випадку виробничі потужності набувають цілком визначені цілочислові значення. Для кожного пункту відома залежність виробничих витрат від випуску, тобто функція $f_i(x_i)$. Для простоти будемо вважати, що $f_i(x_i) = c_i x_i$, тобто функція лінійна. Окрім того, задано п пунктів споживання цього продукту із обсягами споживання $\theta_1$, $\theta_2, \ldots, \theta_n$, а також матриця $C$ транспортних витрат з елементами $c_{ij}$.

Задача полягає у такому розташуванні підприємств, визначенні їхніх виробничих потужностей та організації перевезень, щоб сумарні витрати по виробництву й транспортуванню були мінімальними. На відміну від попередньої постановки задачі додається умова: 4. $x_i$ набуває цілочислові значення.

## 1.3. Постановка задачі математичного програмування

Традиційно термін ``математичне програмування" об'єднує лінійне програмування, цілочислове програмування, випукле та квадратичне програмування, нелінійне програмування й програмування за наявності невизначеностей. Нелінійне програмування має справу з оптимізацією нелінійних функцій із лінійними та (чи) нелінійними обмеженнями. Типовими областями його застосування є прогнозування, планування промислового виробництва, управління товарними ресурсами, контроль якості продукції, що виробляється, проектування технологічних ліній (процесів), облік та планування капіталовкладень.

Метою оптимізації є пошук найкращого розв'язання серед багатьох потенційно можливих відповідно до деякого критерію ефективності або якості. Задача, що допускає лише одне розв'язання, не потребує оптимізації.

Оптимізація може бути здійснена за допомогою багатьох стратегій, починаючи із досить складних аналітичних та чисельних математичних процедур і закінчуючи зваженим застосуванням простої арифметики. Загальні методи оптимізації можна класифікувати таким чином.

1. Аналітичні методи, що використовують класичні методи диференціального та варіаційного числення. Ці методи полягають у визначенні екстремуму функції $f(x)$ шляхом пошуку тих значень $x$, які перетворюють на нуль похідні $df(x)/dx$. Для пошуку екстремуму

функції $f(x)$ за наявності обмежень застосовуються такі методи, як метод невизначених множників Лагранжа, метод обмежених варіацій тощо.

У разі використання аналітичних методів задача оптимізації має бути сформульована математично з метою використання відомих правил для оперування зі всіма функціями та змінними, що є складовими задачі.

2. Чисельні методи, що використовують попередню інформацію з метою побудови розв'язання задачі за допомогою ітеративних процедур. Чисельні методи застосовуються для розв'язання задач, що не можуть бути розв'язані аналітичними методами. Оскільки практичні задачі підлягають розв'язанню чисельними методами, саме вони й є предметом подальших досліджень.

3. Графічні методи, які ґрунтуються на графічному зображенні функції, що підлягає оптимізації. Екстремум функції у цьому випадку одержують безпосередньо шляхом аналізу її графіка. Переваги цього методу полягають у тому, що вони прості й відразу показують, чи існує розв'язання.

4. Експериментальні методи. Екстремум функції можна знайти, виконуючи експерименти безпосередньо із реальними змінними замість того, щоб досліджувати відповідну математичну модель. Результати одного експерименту використовуються для планування наступних експериментів, що дозволяють одержати кращі результати.

5. Методи дослідження різних варіантів. Ці методи ґрунтуються на аналізі кількох можливих розв'язань однієї й тієї ж задачі з метою вибору найкращого. Вибір підходящого засобу опису реальних процесів пов'язаний із рядом труднощів, які з метою зручності обговорення можна розділити на дві групи. Одна група пов'язана із побудовою математичної моделі процесу, друга - із чисельними методами її розв'язання.

Методи нелінійного програмування, як і будь-який математичний апарат, не можна сліпо застосовувати для пошуку розв'язань тієї чи іншої задачі без копіткого попереднього аналізу. Практичне застосування методів нелінійного програмування вимагає від дослідника деякого мистецтва. При цьому вкрай необхідна коректна побудова моделі та застосування відповідних чисельних процедур.

### 1.3.1. Зв'язок задачі нелінійного програмування з реальними процесами

При оптимізації реального процесу параметри та змінні сполучені фізичними законами, такими як закони збереження маси та енергії, що мають бути залучені до математичної моделі як обмеження, незважаючи на те, що вони можуть впливати тільки опосередковано. Таким чином, одна група обмежень складається із функціональних зв'язків, які необхідно враховувати, аби оптимізація була фізично здійсненною. Друга група обмежень містить у собі задані межові діапазони значень змінних або параметрів, що забезпечують їхню фізичну здійсненність або сумісність із даним процесом. До цієї групи належать обмеження у вигляді нерівностей.

Кожне обмеження у вигляді рівностей звужує на одиницю число ступенів свободи в моделі процесу й призводить до появи ще однієї залежної змінної. Кількість ступенів свободи має бути відповідною до кількості незалежних змінних (змінних розв'язання). Це поняття є важливим в задачі оптимізації будь-якого типу, оскільки якщо число змінних дорівнює числу незалежних змінних, то оптимізувати нема чого; значення всіх змінних можна безпосередньо

визначити із сумісного розв'язання системи обмежень у вигляді рівностей $h_i(x) = 0$, $i = \overline{1,n}$. Якщо кількість обмежень $m < n$, єдиний спосіб вирішення задачі - варіювання змінних розв'язання, доки функція мети не досягне свого оптимального значення. Якщо ж кількість обмежень перевищує кількість незалежних змінних, функція мети має представляти собою деякий статистичний критерій, наприклад, такий, як у методі найменших квадратів.

### 1.3.2. Випуклість

Випуклість є одним із найважливіших математичних понять, що використовуються в математичному програмуванні. Визначимо спочатку поняття випуклої множини. Множина $W \subset E^n$ називається випуклою, якщо з того, що $x, yinW$ випликає $(1-\lambda)x + \lambda y \in W$ за будь-якого $\lambda \in [0,1]$. Геометрично це означає, що якщо дві точки належать множині $W$, то й увесь відрізок, що сполучує ці точки, також належить множині $W$. Прикладами випуклих множин можуть слугувати відрізок, промінь, пряма, многогранні множини, що наявні у задачах лінійного програмування, $n$-вимірна куля (відкрита чи замкнена); $n$-вимірна сфера $\| x - x_0 \| = r$ не є випуклою множиною.

Кожна задача математичного програмування має свою допустиму область у просторі стану яка складається з усіх точок, що задовольняє всім обмеженням задачі.

Аналогічно можна визначити поняття випуклої функції. Функція $f(x)$ є випуклою на випуклій множині $S$, якщо для будь-яких двох точок $x_1, x_2 \in S$ і для всіх скалярів $\lambda$, $0 \le \lambda \le 1$

$$f[\lambda x_1 + (1-\lambda)x_2] \le f(x_1) + (1-\lambda) f(x_2). \qquad (1)$$

Якщо в (1) буде строга нерівність, функція $f(x)$ строго випукла. Якщо знак нерівності змінити на протилежний, $f(x)$ -- угнута функція.

Випукла функція має глобальний мінімум, тоді як невипукла функція може мати декілька локальних мінімумів. Ця властивість має особливе значення для більшості алгоритмів чисельних розв'язань задач мінімізації.

### 1.3.3. Теорема Куна-Такера

Однією з найважливіших теорем математичного програмування є теорема Куна-Такера. Припустимо, що задача математичного програмування сформульована таким чином:

мінімізувати $f(x)$

$$\text{з умовою } g_i(x) \le 0,\ j = \overline{1,m},\ x_i \ge 0,\ j = \overline{1,m}, \qquad (2)$$

де $f(x)$, $g_i(x)$ ( $j = \overline{1,m}$ ) -- випуклі функції $n$ змінних $x$.

Запровадимо сукупність із $m$ змінних $u_1, u_2, \ldots, u_m$, які зазвичай мають назву *множників Лагранжа* й утворюють вектор $u$. Запровадимо нову функцію $L(x,u)$ -- функцію Лагранжа таким чином:

$$L(x,u) = f(x) + \sum_{j=1}^{m} u_j g_j(x). \qquad (3)$$

Сформулюємо також таку умову:

знайдеться точка $x^*$ така, що $x_i^* \geq 0$, $g_j(x) < 0$, $j = \overline{1,m}$. Ця умова має назву *умови Слейтера*.

**Теорема Куна-Такера**.

Нехай виконується *умова Слейтера* (у разі лінійних обмежень $g_j(x)$) ця умова може бути відкинута). Вектор $x^0$ є розв'язанням задачі оптимізації (3) тоді і тільки тоді, коли існує такий вектор $u^0$, що при всіх $x_i \geq 0$, $u_j \geq 0$

$$\begin{aligned} & x_i^0 \geq 0, \qquad i = \overline{1,n}, \\ & u_j^0 \geq 0, \qquad j = \overline{1,m}, \\ & L(x^0,u) \leq L(x^0,u^0) \leq L(x,u^0). \end{aligned} \tag{4}$$

має глобальний мінімум в області $x_i \geq 0$ при $x = x^0$. За фіксованого $x^0$ вона має глобальний максимум в області $u_j \geq 0$ при $u = u^0$. Екстремальна точка ( $x^9u^0$ ) із цими властивостями зветься *сідловою точкою*. З цієї причини теорему Куна--Такера часто називають теоремою про сідлову точку.

Не зупиняючись на доведенні цієї теореми, можна навести такі евристичні міркування, що підтверджують теорему. Якщо замістити (3) у (4), одержимо для всіх $x_i \geq 0$, $u_j \geq 0$:

$$f(x^0) + \sum_{j=1}^{m} u_j g_j(x^0) \leq u_j^0 g_j(x^0) \leq f(x) + \sum_{j=1}^{m} u_j^0 g_j(x). \tag{5}$$

З лівої нерівності цієї формули витікає, що

$$\sum_{j=1}^{m} u_j g_j(x^0) \leq u_j^0 g_j(x^0). \tag{6}$$

Оскільки (6) справджується за всіх $u_j \geq 0$, то

$$g_j(x) \leq 0, \tag{7}$$

і

$$\sum_{j=1}^{m} u_j^0 g_j(x^0) = 0. \tag{8}$$

Із нерівності (7) випливає, що точка $x^0$ -- допустима точка. Зважаючи на (8), праву частину нерівності (5) можна записати так:

$$f(x^0) \leq f(x) + \sum_{j=1}^{m} u_j^0 g_j(x) \text{ çàâñ³õ} x_i \geq 0. \tag{9}$$

Згідно з вимогами допустимості $u_j^0 \geq 0$, $g_j(x) \leq 0$, тоді з (8) випливає, що

$$f(x^0) \leq f(x), \tag{10}$$

Це означає, що $f(x^0)$ дійсно є мінімум $f(x)$ у допустимій області і $x^0$ дійсно є розв'язання задачі (3).

У випадку диференційовних функцій умови Куна - Такера можна записати у такому вигляді.

$$\frac{\partial L}{\partial x_i}\Big|_{x^0,u^0} \geq 0, \quad x_i^0 \frac{\partial L}{\partial x_i}\Big|_{x^0,u^0} = 0,\ x_i^0 \geq 0,\ i = \overline{1,n}; \tag{11}$$

$$\frac{\partial L}{\partial u_j}|_{x^0,u^0} \le 0, \quad u_j^0 \frac{\partial L}{\partial u_j}|_{x^0,u^0} = 0, u_i^0 \ge 0, \ j = \overline{1,m}. \tag{12}$$

### 1.3.4. Двоїстість

Поняття двоїстості відіграє важливу роль у математичному програмуванні. Розглянемо таке формулювання задачі мінімізації:

$$(P) = \min_x \{ f(x) \mid g_j(x) \le 0, \ j = \overline{1,m} \}. \tag{13}$$

Ця задача має назву прямої задачі (Р). Існує сполучена з цією задача максимізації, що має назву *двоїстої* $(D)$ або сполученої із прямою задачею. Двоїста задача формулюється так:

$$(D) = \max_x \left\{ L(x,u) \mid \frac{\partial L}{\partial x_i} = 0, i = \overline{1,n}, \quad u_j \ge 0, \ j = \overline{1,m} \right\}. \tag{14}$$

де $L(x,u)$ -- функція, що визначається формулою (3).

Поняття двоїстості встановлює визначені співвідношення між розв'язаннями прямої задачі та двоїстої задачі. Зв'язок між функціями мети $f(x)$ та $L(x,u)$ двох задач визначає така теорема.

**Теорема**. *Якщо* $f(x)$ *і* $g_j(x)$ *(* $j = \overline{1,m}$*) є випуклими диференційовними функціями, у є будь-яка допустима точка прямої задачі (13) та якщо* $(x^f, u^f)$ *позначає будь-яку допустиму точку двоїстої задачі (14), то*

$$f(y) \le L(x^f, u^f). \tag{15}$$

Практичне значення цієї теореми полягає у тому, що якщо алгоритми, що розв'язують пряму задачу (Р), дають двоїсті допустимі точки, то це дозволяє одержати оцінку знизу для оптимального значення $(y)$.

## 1.4. Загальна постановка задач прийняття рішення

У теорії прийняття рішень процес прийняття рішення розглядається як процес перетворення інформації, що включає етапи змістовної постановки задачі, формування переліку можливих дій (альтернативних рішень) й їхніх наслідків, вибір альтернативного рішення, аналіз результатів прийнятого рішення. Часто такий процес носить ітеративний характер, наприклад, у результаті отриманого рішення може бути скоригована вихідна множина альтернативних рішень.

У загальному вигляді задача прийняття рішення (ЗПР) може бути сформульована в термінах умов - мети. Умови включають множину станів об'єкта й множину операторів, що переводять об'єкт із одного стану в інший. Інакше, завдання умов представляється формальним описом засобів, результатів та способу їхнього зв'язку. Мета ж визначає бажаний стан об'єкта. Різноманіття практичних задач прийняття рішень обумовлює різноманіття деталізацій цієї загальної постановки.

Запровадимо множини $W$ та $U$ ; елементи множини $W$ будемо називати альтернативними рішеннями чи просто альтернативами, елементи множини $U$ --

результатами. Тоді задача прийняття рішення полягає у виборі альтернативи $xOW$, що призводить до деякого результату $uOU$. Ефективність рішення визначається ступенем відповідності отриманого результату поставленій меті.

Формалізація цієї постановки допускає виконання таких етапів: формалізацію мети, побудову формального опису деякої універсальної множини альтернативних рішень $Wu$ (множини всіх можливих у різних реалізаціях даної ЗПР альтернативних рішень), аналіз цієї множини та виділення множини допустимих (з урахуванням обмежень конкретної реалізації ЗПР) альтернатив $WHWu$, побудова формального опису множини результатів $U$ й взаємозв'язку $W$ та $U$ -й, нарешті, побудова самої формальної процедури вибору $OW$.

У різних моделях ТПР під задачею прийняття рішення часто мається на увазі тільки сам акт вибору $x$ із заданого $W$. Таке обмеження проблеми може бути виправдано тим, що в деяких практичних задачах множини $U$ та $W$ фіксовані, але незважаючи на це, проблема прийняття рішення залишається складною. З іншого боку, у різних складних задачах прийняття рішень задача вибору $xOW$ (або послідовність таких задач) завжди має місце й є центральною.

Проте звернемо увагу на той факт, що процес вироблення альтернатив має першорядне значення, оскільки успішне його завершення гарантує, що всі ``гарні'' альтернативи будуть брати участь у виборі. Тому автоматизація процесів прийняття рішень має охоплювати всі етапи процесу прийняття рішення, а відповідні програмно-алгоритмічні засоби мають включати процедури підтримки цих етапів. У процесі ухвалення рішення беруть участь:

1. особа, яка приймає рішення (ОПР);
2. експерти;
3. консультанти.

Особою, яка приймає рішення, називають людину, яка має мету, що слугує мотивом постановки задачі та пошуку її вирішення; людину, що є компетентним фахівцем у своїй галузі, відповідно до уявлень якого про проблемну ситуацію здійснюється її формалізація; людину, яка наділена необхідними повноваженнями, відповідно до яких вона приймає остаточне або проміжне рішення й несе повну або часткову відповідальність за його наслідки.

При формалізації постановки ЗПР використовується одержувана від ОПР інформація про його надання переваг. Під наданням переваг ОПР будемо розуміти сукупність зазвичай спочатку не структурованих його представлень, пов'язаних з перевагами та недоліками різних альтернативних рішень (тобто має місце деяка вихідна невизначеність надання переваг). Інформація про надання переваг використовується на різних етапах формалізації ЗПР та її вирішення.

Експертом називають фахівця, який може надавати інформацію, необхідну для формалізації ЗПР, але не несе відповідальність за наслідки прийнятого рішення. Одержувана від експертів інформація часто використовується для опису $W, U$ та їхнього взаємозв'язку.

Консультантом (дослідником, системним аналітиком) називають фахівця з теорії прийняття рішень, що здійснює формалізацію ЗПР, розробляє процедуру прийняття рішень, організує роботу експертів та ОПР.

Іноді під ОПР мається на увазі не єдина людина, а група людей, які виконують визначені функції в процесі прийняття рішень. За типом їхньої участі в процесі ухвалення рішення можуть бути відокремлені такі класи задач:

1. Індивідуальне прийняття рішень. На різних етапах прийняття рішення бере участь єдиний ОПР, він же несе повну відповідальність за прийняте рішення.

2. Колективне прийняття рішень. У цьому випадку має місце багатоетапний вибір. На

різних етапах здійснюється вибір, що звужує допустиму множину альтернатив $W$ різними особами. Остаточний вибір здійснюється єдиною особою, відповідно до її мети.

3. Групове прийняття рішень. Тут ставиться задача вироблення узгодженого рішення на основі індивідуальних виборів багатьох осіб, що додержуються власних цілей. У цьому випадку немає ``головного ОПР'', а потрібно знайти ``найбільш справедливе'' рішення, ``щонайкраще'', що задовольняє цілі (часто суперечливі) різних ОПР.

# Розділ 2. Лінійне програмування

## 2.1. Задача лінійного програмування

Задачею лінійного програмування зветься задача, в якій функція мети є лінійна функція, і на незалежні змінні накладаються лінійні обмеження.

Комбінація скалярів та векторів, що зазвичай позначається як $c_i x_i$ , називається лінійною, якщо вона може бути записана у вигляді

$$c_1 x_1 + c_2 x_2 + \cdots + c_n x_n, \tag{16}$$

де всі коефіцієнти $c_i$ - сталі. Наприклад, функція

$$4x_1 + 3x_2 + 3x_3 + 2$$

є лінійною відносно змінних $x_1, x_2, x_3$, тоді як функція

$$2x_1^2 + x_1 x_2 + 3x_2^2$$

є нелінійною відносно тих же змінних. Величини $x_1, \ldots, x_n$ називаються лінійно залежними, якщо для деякого набору $c_i$ має місце рівність:

$$c_1 x_1 + c_2 x_2 + \cdots + c_n x_n = \sum_{k=1}^{n} c_k x_k = 0. \tag{17}$$

З іншого боку, якщо $\sum_{k=1}^{n} c_k x_k = 0$ тільки тоді, коли всі коефіцієнти $c_k = 0$ , $x_1, \ldots, x_n$ називаються лінійно незалежними.

Задача лінійного програмування може бути сформульована у такому вигляді.

Мінімізувати

$$y = \sum_{k=1}^{n} c_k x_k \tag{18}$$

з обмеженнями

$$\sum_{k=1}^{n} a_{kj} x_k = b_j \geq 0, \; j = \overline{1, m}. \tag{19}$$

$$x_k \geq 0, \; k = \overline{1, n}, \tag{20}$$

де $a_{kj}, b_j$ і сталі, а $x$ -- вектор шуканих змінних Для розв'язання задачі (3)-(5) було запропоновано різноманітні методи.

Розглянемо таку задачу лінійного програмування:

максимізувати $y = x_1 + 4x_2$ за умови

$$x_1 \leq 10, \; x_2 \leq 10,$$

$$x_1 + x_2 \leq 14, \quad x_1 \geq 0, \; x_2 \geq 0.$$

Аналіз обмежень показує, що допустима область обмежена многокутником (рис. 2.1) ABCDO. Прямі $L_1$ , $L_2$ , $L_3$ відповідають сталим значенням функції мети по мірі їхнього зростання. Вочевидь, пряма $L_2$, для якої $c^T x = 44$, відповідає максимуму функції мети, де ця пряма має як мінімум одну точку $B$ , що є спільна із допустимою областю. Значить,

оптимальним розв'язанням задачі буде $x_{1,0} = 4$ та $x_{2,0} = 10$.

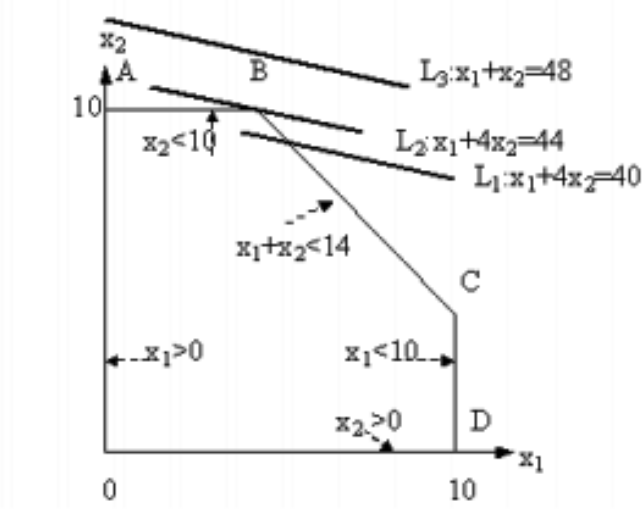


Рис.2.1 Допустима область для задачі ЛП

Інша, загальніша постановка задачі лінійного програмування може бути сформульована таким чином.

Дана функція мети

$$f(x) = \sum_{i=1}^{N} c_i x_i = c^T x. \tag{21}$$

Треба знайти значення змінних $x_i \geq 0$, $i = \overline{1 < N}$, що задовольняють таким обмеженням у формі лінійних рівностей та нерівностей:

$$\sum_{k=1}^{N} a_{ik} x_k \leq b_i, \ i = \overline{1, p}, \tag{22}$$

$$\sum_{k=1}^{N} a_{ik} x_k = b_i, \ i = \overline{p+1, p+q}, \tag{23}$$

$$\sum_{k=1}^{N} a_{ik} x_k \geq b_i, \ i = \overline{p+q+1, m}. \tag{24}$$

Тобто, задано $q$ обмежень у формі рівностей та $m - q$ обмежень у формі нерівностей.

Оскільки у загальному випадку легше оперувати із рівностями, ніж із нерівностями, то (7) та (9) перетворимо у рівності шляхом додавання додаткових змінних.

Для нерівності вигляду (7) додаткові змінні $x_{li}$ запроваджуються так:

$$\sum_{k=1}^{N} a_{ik} x_k + x_{li} = b_i, \ i = \overline{1, p}. \tag{25}$$

Для обмежень вигляду (9) додаткові змінні $x_{li} \geq 0$ запроваджуються так:

$$\sum_{k=1}^{N} a_{ik} x_k - x_{li} = b_i, \ i = \overline{p+q+1, m}. \tag{26}$$

Слід зауважити, що в функцію мети додаткові змінні не входять й не впливають на оптимальне розв'язання. Додаткові змінні використовуються з метою зведення обмежень до

такої стандартної форми:

$$Ax = b, \qquad (27)$$

де $x \geq 0$ тепер представляють $n = (N + m - q)$-вимірний вектор, що містить у собі всі основні та додаткові змінні, $A$ - стала матриця розмірності $m \times n$.

Отже, (12) являє собою систему $m$ лінійних рівнянь із $n$ невідомими $n \geq m$.

Задача лінійного програмування може бути тепер сформульована у такій, класичній формі.

$$\max(\min)\{c^T x \; Ax = b, \; x_i \geq 0, \; i = \overline{1, N}\}, \qquad (28)$$

де $c^T = [c_1, c_2, \ldots, c_N, 0, \ldots, 0]$ -- $n$-вимірний вектор-рядок, $N$ - кількість основних змінних.

*Приклад.*

Привести до канонічного (стандартного) вигляду задачу

$$\begin{aligned}
&3x_1 - 2x_2 + x_3 && \to \max;\\
&2x_1 - 3x_3 && \leq -1;\\
&3x_1 + 4x_2 + 2x_3 && = 6;\\
&x_1 + 4x_2 - x_3 && \leq 2;\\
&x_i \geq 0, && j = 1, 2, 3.
\end{aligned}$$

Щоб привести задачу до еквівалентного канонічного вигляду запровадимо у першому та третьому обмеженнях дві додаткові змінні $x_4, x_5$. Отримуємо

$$\begin{aligned}
&3x_1 - 2x_2 + x_3 && \to \max;\\
&2x_1 - 3x_3 + x_4 && = -1;\\
&3x_1 + 4x_2 + 2x_3 && = 6;\\
&x_1 + 4x_2 - x_3 + x_5 && = 2;\\
&x_i \geq 0, && j = 1, 2, 3, 4, 5.
\end{aligned}$$

Геометричний зміст задачі лінійного програмування у випадку двох змінних неважко зрозуміти, якщо розв'язати цю задачу шляхом геометричних побудов. Нехай дана задача

$$\begin{aligned}
&c_1 x_1 + c_2 x_2 && \to \max;\\
&a_{11} x_1 + a_{12} x_2 && \leq b_1,\\
&a_{21} x_1 + a_{22} x_2 && \leq b_2,\\
&\ldots + \ldots && \ldots\\
&a_{m,1} x_1 + a_{m,2} x_2 && \leq b_m,
\end{aligned}$$

Запровадимо на площині декартову систему координат й зіставимо кожній парі чисел $(x_1, x_2)$ точку площини із координатами $x_1$ та $x_2$. З'ясуємо спочатку якою буде множина точок, що відповідає допустимим розв'язкам даної задачі. Для цього розглянемо таку лінійну нерівність

$$a_{11} x_1 + a_{12} x_2 \leq b_1.$$

Вона, як відомо, визначає на площині одну з двох півплощин, на які пряма $a_{11} x_1 + a_{12} x_2 \leq b_1$ поділяє площину. При цьому відповідна півплощина містить і межову пряму (замкнута півплощина). З метою визначення шуканої півплощини замістимо у даній нерівності координати однієї деякої точки, що не належить межовій прямій. Якщо при цьому нерівність

справджується, то це й буде шуканою півплощиною.

Отже, кожне з обмежень задачі (13) задає на площині деяку півплощину. Нас цікавлять точки площини, координати яких задовольняють всім нерівностям- обмеженням, тобто точки, що належать одночасно всім $m$ півплощинам, які визначаються окремими обмеженнями. Отже, допустима множина задачі лінійного програмування геометрично зображується перетином (загальною частиною) півплощин, що визначаються окремими обмеженнями. Будемо називати цей перетин *допустимою областю* задачі лінійного програмування.

Якщо допустима область не порожня, вона являє собою деякий многокутник (що може бути й необмеженим). Умовно будемо називати многокутниками допустимі області, що складаються з одного відрізку, смуги або точки. Вочевидь, в усіх випадках відрізок, що з'єднує будь-які дві точки допустимої області, цілком міститься в цій області. Області з такою властивістю називаються *випуклими*.

Отже, якщо допустима область задачі лінійного програмування не порожня, то вона випукла. Нехай допустима область задачі лінійного програмування виявилась не порожньою. Оптимальними є ті точки допустимої області, координати яких доставляють функції мети найбільше значення. Функція $c_1x_1 + c_2x_2$ набуває одне й те саме значення $a$ в усіх точках прямої $c_1x_1 + c_2x_2 = a$ де $a$ - деяке дійсне число. Якщо вважати $a$ параметром, тримаємо сімейство паралельних прямих, що називаються лініями сталого значення, або *лініями рівня* функції $f(x_1, x_2) = c_1x_1 + c_2x_2$. Нас цікавлять ті точки допустимої області, які належать лінії рівня із найбільшим значенням $a$ порівняно із його значеннями для усіх інших ліній рівня, що перетинаються із допустимою областю. Бачимо, що значення $a$ (а значить, і значення функції $c_1x_1 + c_2x_2$ зростають необмежено, якщо переміщувати пряму $c_1x_1 + c_2x_2 = a$ у напрямку її нормалі - вектора $c = (-c_1; -c_2)$.

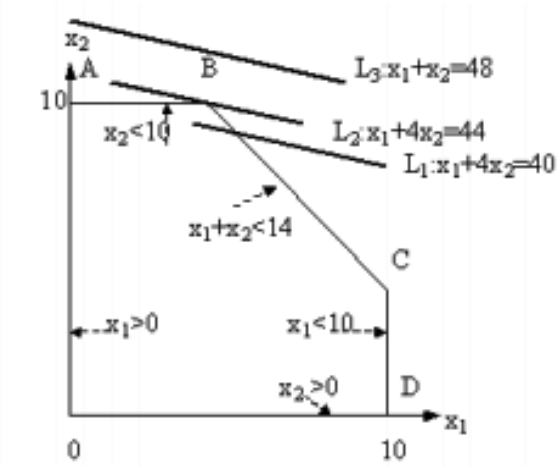


Рис.2.2. Визначення кутової точки в задачі ЛП

Отже, щоб відшукати оптимальні точки допустимої множини задачі на максимум, слід переміщувати пряму $c_1x_1 + c_2x_2 = a$ у напрямку вектора $c = (-c_1; -c_2)$, починаючи з деякого фіксованого положення, за яким вона перетинається із допустимою областю (точка $A$ на рис. 2.2) доти, доки вона не перестане перетинатись із цієї областю. Перетин допустимої області із лінією рівня у тому її положенні, коли подальше переміщення дає порожній перетин, й буде

множиною оптимальних точок задачі лінійного програмування. На рис. 2.2 це єдина точка $P(\overline{x}_1, \overline{x}_2)$. У випадку задачі мінімізації лінію рівня треба переміщувати у протилежному напрямку.

З точки зору практики важливими є випадки, коли задача лінійного програмування має оптимальні розв'язання. При $n = 2$ очевидно, що задача лінійного програмування має оптимальні розв'язання, якщо її допустима область не порожня та обмежена (тобто цілком розташована у крузі з центром у початку координат досить великого радіусу).

Таким чином, для $n = 2$ геометрично очевидна така необхідна і достатня умова існування оптимального розв'язання.

Задача лінійного програмування на максимум (мінімум) тоді і тільки тоді має оптимальне розв'язання, коли її функція мети обмежена зверху (знизу) у допустимій області.

Якщо виконати аналогічний аналіз для випадку $n = 3$, тобто задачі вигляду

$$\begin{aligned} & c_1x_1 + c_2x_2 + c_3x_3 && \to \max(\min); \\ & a_{11}x_1 + a_{12}x_2 + a_{13}x_3 && \le b_1, \\ & a_{21}x_1 + a_{22}x_2 + a_{23}x_3 && \le b_2, \\ & \ldots\ldots\ldots\ldots\ldots\ldots && \ldots \\ & a_{m,1}x_1 + a_{m,2}x_2 + a_{m,3}x_3 && \le b_m, \end{aligned}$$

ми переконаємось, що сумісна система лінійних нерівностей із трьома змінними визначає в тривимірному просторі деякий многогранник; значення функції мети збільшуються у напрямку вектора $c = (c_1; c_2; c_3)$ і зменшуються у протилежному напрямку; якщо переміщувати площину рівня $c_1x_1 + c_2x_2 + c_3x_3$, починаючи з деякого фіксованого положення (доки вона перетинається із допустимим многогранником) у напрямку вектора $c = (c_1; c_2; c_3)$ для задачі на максимум або у напрямку $-c = (-c_1; -c_2; -c_3)$ для задачі на мінімум, точки перетину із допустимою областю у тому її крайньому положенні, коли цей перетин ще не порожній, визначає оптимальне розв'язання даної задачі.

Якщо прийняти до уваги, що поняття та властивості $n$-вимірного простору є результат формалізації та узагальнення понять та властивостей звичного тривимірного евклідового простору, стане ясно, що вони мають виконуватись для задачі лінійного програмування й у загальному випадку.

## 2.2. Симплекс-метод

Симплекс - метод, запропонований Дж. Данцігом, є одним із найбільш ефективних алгоритмів чисельного розв'язання задач лінійного програмування. Зауважимо, що будь-який вектор $x$, компоненти якого задовольняють системі рівнянь ((12)), являє собою вектор допустимих розв'язань задачі. У симплекс-методі є дуже важлива особлива форма допустимого розв'язання -- базисне допустиме розв'язання.

### 2.2.1. Визначення базисного розв'язання

Нехай дано систему т лінійних рівнянь вигляду ((12)) з $n$ невідомими. Припустимо, що матриця $A$ має ранг $m$. Базисом зветься набір з змінних таких, що матриця коефіцієнтів

при цих змінних у $m$ обмеженнях є неособлива. Ці $m$ змінних називаються базисними змінними (по відношенню до даного базису); решта змінних називаються небазисними змінними. Якщо покласти всі небазисні змінні рівними нулю й розв'язати обмеження відносно базисних змінних, отримаємо базисне розв'язання, сполучене з цим базисом. Якщо базисне розв'язання є допустимим, тобто

$$x_i \geq 0,\ i = \overline{1,n}, \tag{29}$$

його називають допустимим базисним розв 'язанням.

Розглянемо тепер геометричну структуру, що визначається формулою ((12)). Кожне з $m$ рівнянь визначає гіперплощину в $n$-вимірному просторі. Вочевидь, гіперплощина являє випуклу множину. припустимо, що точки $x_1$, $x_2$ містяться у деякій $i$-й гіперплощині. Тоді

$$a^T x_1 = 0 \tag{30}$$

та

$$a^T x_2 = 0 \tag{31}$$

також належить цій гіперплощині, оскільки

$$a^T x = a_i[\lambda x_2 + (1-\lambda)x_1 = \lambda b_i + (1-\lambda)b_i = b_i. \tag{32}$$

Відкритий і замкнутий півпростори, обмежені гіперплощиною, також є випуклими множинами. Замкнені півпростори, що визначаються нерівностями $x_i \geq 0$, також є випуклі. Таким чином, множина допустимих розв'язань задачі ((13)) випукла.

Запровадимо поняття крайніх (або екстремальних) точок випуклих множин.

**Визначення крайньої точки.**

Точка $x$ є крайня точка випуклої множини, якщо не існує інших точок $x_1$, $x_2$, що належать даній множині таких, що

$$x = \lambda x_2 + (1-\lambda)x_1,\ 0 < \lambda < 1.$$

Отже, крайня точка не може належати відрізку, що з'єднує будь-які дві інші точки цієї множини. Крайня точка фактично є ``кутовою точкою" випуклої множини.

**Теорема**. *Кожне допустиме базисне розв'язання системи рівнянь (12) є крайня (екстремальна) точка випуклої множини й, навпаки, кожна крайня точка множини (12), (14) є допустиме базисне розв'язання*.

Симплексний алгоритм визначається такими кроками:

1. Знайти будь-яке допустиме базисне розв'язання рівностей-обмежень (12).

2. Перейти від першого допустимого розв'язання до наступного базисного розв'язання у напрямку, за яким значення функції мети зростає (спадає).

3. Продовжувати переходити від одного допустимого базисного розв'язання до іншого, доки не буде досягнуто найбільше (найменше) значення функції мети.

В системі з $m$ рівнянь із $n$ невідомими ($n > m$) загальна кількість базисних розв'язань дорівнює

$$C_n^m = \frac{n!}{m!(n-m)!}. \tag{33}$$

Кандидатами на оптимальне розв'язання задачі лінійного програмування є крайні точки допустимої області. Ітерації симплексного алгоритму являють собою перехід від однієї крайньої точки до іншої у напрямку зростаючого (спадаючого) значення функції мети.

Сукупність крайніх точок області допустимих розв'язань називається *опорним розв'язанням*.

Вочевидь, оптимальним розв'язанням буде одна з крайніх точок, у якій функція мети приймає екстремальне значення.

Симплекс-метод забезпечує розв'язання задач лінійного програмування, що сформульовано у канонічній (стандартній) формі, коли обмеження являють собою рівності.

Змінні у канонічній формі запису задачі лінійного програмування поділяються на базисні (т $m$ змінних) та вільні (небазисні) ( $k = N - m = n + m - m = n$ змінних). Базисні змінні та функція мети мають бути виражені через вільні змінні. При цьому слід відокремлювати вільні змінні та основні, й базисні та додаткові.

Якщо прийняти $m$ додаткових змінних як базисні та виразити їх через $k = n$ основних змінних, тримаємо перше базисне розв'язання. У першому базисному розв'язанні всі основні змінні є вільними й дорівнюють нулю, всі додаткові - базисними і дорівнюють правим частинам системи обмежень. Після визначення першого базисного розв'язання задачу лінійного програмування можна записати у вигляді стандартної таблиці.

### 2.2.2. Пошук опорного розв'язання

Подання умов задачі лінійного програмування у вигляді стандартної таблиці надає можливість виявити несумісність обмежень та необмеженість функції мети. Відсутність області допустимих розв'язань (несумісність обмежень) виявляється на кожній ітерації пошуку опорного розв'язання.

**Ознака 1**. Обмеження несумісні, якщо на кожній ітерації пошуку опорного розв'язання у кожному рядку, що має від'ємний вільний член, немає жодного від'ємного елементу.

**Ознака 2**. Функція мети не буде обмеженою знизу (зверху), тобто відсутній min (max) функції, якщо у кожному стовпці, в якому у рядку коефіцієнтів функції мети міститься додатний (від'ємний) елемент, немає жодного від'ємного (додатного) елемента.

Після перевірки за наведеними ознаками визначається опорне розв'язання.

**Ознака 3**. Розв'язання буде опорним, якщо в стандартній таблиці всі вільні члени невід'ємні.

Основу алгоритму визначення опорного розв'язання складає метод модифікованих жорданових виключень.

1. Стандартна таблиця заповнюється на основі стандартної форми запису задачі.

2. При виборі розв'язуючого елементу розглядаються для обміну стовпці та рядки, що містять коефіцієнти при змінних.

а). Обирається стовпець, що розв'язує. Для цього у рядку, що містить вільний член, обирається від'ємний елемент та стовпець, що містить цей елемент, обирається як розв'язуючий. Якщо у рядку, що містить від'ємний вільний член, відсутній від'ємний елемент, це означає, що не виконується ознака 1, тобто обмеження несумісні.

Якщо у рядку, що містить від'ємний вільний член, є кілька від'ємних елементів, можна обирати будь-який з них. При цьому як опорне розв'язання може бути отримано різні вершини.

б). Обирається розв'язуючий рядок. В розв'язуючому стовпці розглядаються всі елементи, які мають знак, однаковий зі знаком вільного члену того ж рядка. Як розв'язуючий рядок обирається той рядок, для якого відношення вільного члену до елементу розв'язуючого стовпця є мінімальним.

$$\min_{i\le m}\left\{\frac{b_i}{a_{ij}}\right\},$$

в). Виділяється розв'язуючий елемент $a_{ij}$.

3. Нижню праву частину вічка елемента, що розв'язує, заповнюють значенням А = $\lambda = 1/a_{ij}$. Нижні праві частини вічок рядку, що розв'язує, - добутками елементів, що містяться в лівих верхніх частинах кожного вічка, на $\lambda$, а стовпця, що розв'язує - добутками елементів, що містяться в лівих частинах кожного вічка, на $-\lambda$.

Виділяються (крім елемента, що розв'язує) в розв'язуючому рядку всі верхні числа, а в стовпці, що розв'язує, всі нижні. Нижні праві частини решти вічок стандартної таблиці заповнюються добутками цих чисел, що містяться в тому ж стовпці та того ж рядка, що й вічко, яке заповнюється.

4. Скласти наступну стандартну таблицю, в якій поміняти місцями змінні $x_i$, $x_{n+i}$, решту змінних залишити на своїх місцях, записати у верхні ліві частини вічок елементи рядку й стовпця, що розв'язують, які містяться в нижніх правих частинах тих же вічок попередньої таблиці (для стовпця, що розв'язує, це співпадає із елементами, які вже виділено); у верхніх лівих кутах вічок, що залишились, записати суму двох величин, що містяться в кожному вічку попередньої таблиці.

5. У процесі пошуку опорного розв'язання необхідно перевіряти виконання ознаки 3. Якщо вона виконується, опорне розв'язання знайдено і треба переходити до пошуку оптимального розв'язання. Якщо ні - перевіряються виконання ознак 1 та 2. Якщо вони не виконуються, ітерацію обміну змінних слід повторити, якщо да - задача не має розв'язання.

### 2.2.3. Симплекс-таблиця

Розглянемо тепер таку задачу лінійного програмування.

$$\max\left\{ f(x)=\sum_{i=1}^{N}c_i x_i \mid \sum_{i=1}^{N}a_{ji}x_i \le b_j;\ b_j\ge 0,\ j=\overline{1,m},\ x_i\ge 0,\ i=\overline{1,n}\right\}. \tag{34}$$

Запровадимо $m$ додаткових змінних $x_{N+1}$, …, $x_{N+m}$ і перетворимо обмеження у формі нерівностей до вигляду

$$\sum_{i=1}^{N}\sum_{k=1}^{N}a_{jk}x_k + x_{N+j} = b_j,\ x_{n+j}\ge 0,\ j=\overline{1,m}. \tag{35}$$

або в матричній формі

$$Ax = b. \tag{36}$$

Матриця $A$ складається з таких $n = N + m$ елементів

$$\left[\begin{array}{cccc|cccc} a_{1,1} & a_{1,2} & \dots & a_{1,N} & 1 & 0 & \dots & 0 \\ a_{2,1} & a_{2,2} & \dots & a_{2,N} & 0 & 1 & \dots & 0 \\ \dots & & & & & & & \\ a_{N,1} & a_{N,2} & \dots & a_{N,N} & 0 & 0 & \dots & 1 \end{array}\right] \tag{37}$$

$$x = [x_1, x_2, \dots, x_N, x_{N+1}, \dots, x_{N+m}]^T. \tag{38}$$

Аналіз (19) свідчить, що розв'язання

$$x_1 = x_2 = \cdots = x_N = 0; \tag{39}$$

$$x_{N+i} = b_i,\ i = \overline{1,m} \tag{40}$$

є допустимим базисним розв'язанням. Значить, з нього можна починати пошук розв'язання задачі лінійного програмування. Згідно з (20) можна записати для базисних змінних

$$x_{N+j} = b_j - \sum_{k=1}^{N} a_{jk} x_k \geq 0,\ j = \overline{1,m} \tag{41}$$

Вектор $c$ функції мети доповнимо нульовими елементами, що відповідають змінним $x_{N+j}$. Новий вектор

$$c^T = [x_1, x_2, \ldots, c_N, c_{N+1}, \ldots, c_{N+m}], \tag{42}$$

$$c_{N+1} = c_{N+2} = \ldots = c_{N+m} = 0.$$

Значення функції мети при допустимому базисному розв'язанні (23), (25) дорівнює

$$f_0(x) = \sum_{j=1}^{m} c_{N+j} x_{N+j} = \sum_{j=1}^{m} c_{N+j} b_j = 0 \tag{43}$$

Враховуючи (27), функцію мети можна записати у вигляді

$$f(x) = \sum_{i=1}^{m} c_i x_i = f_0(x) + \sum_{j=1}^{m} \left[ c_i - \sum_{j=1}^{m} c_{N+j} a_{ji} \right] x_i, \tag{44}$$

або

$$f(x) - f_0(x) = \sum_{i=1}^{m} z_i x_i. \tag{45}$$

де

$$z_i = c_i - \sum_{i=1}^{m} c_{N+j} a_{ji} = c_i. \tag{46}$$

Припустимо тепер, що виконується наступна корекція: значення однієї з небазисних змінних $x_i$, $i = \overline{1,N}$ дорівнює 1 замість нуля. Ненульові базисні змінні змінюються таким чином, що розв'язання залишається допустимим, тобто усі обмеження виконуються. Тоді з (30) випливає

$$F(x - f_0(x) = z_i,\ i = \overline{1,N}. \tag{47}$$

Отже, $z_i$ є величина, на яку змінюється функція $f(x)$ порівняно з її значенням при допустимому базисному розв'язанні у випадку виконання корекції, що описана вище. Якщо $z_i > 0$, значення $f(x)$ внаслідок корекції збільшується. Звідси витікає правило: корекція виконується доти, доки всі $z_i$ не виявляться від'ємними. Якщо виявиться, що додатними виявились більше однієї $z_i$, для наступної ітерації обирають координату із найбільшим за абсолютною величиною значенням:

$$\max |\, z_i \,|,\ z_i > 0. \tag{48}$$

Слід звернути увагу на той факт, що у випадку

$$a_{ji} \leq 0 \tag{49}$$

для всіх $j$ задача лінійного програмування не має розв'язання.

Максимальне значення $f(x)$ буде досягнуто, коли перше із $x_{N+j}$ перетвориться у нуль. Ця змінна й виводиться з базису. Її номер $r$ визначається відповідно до за умови

$$\frac{b_j}{a_{ri}} = \min\left\{\frac{b_j}{a_{ji}}, a_{ji} > 0\right\}. \tag{50}$$

Елемент $a_{ri}$ симплексної таблиці, що міститься на перетині $r$-го рядка та $i$-го стовпця, називається ведучим (або розв'язуючим) елементом. При цьому $r$-й рядок, який відповідає змінній, що виключається із базису, називається рядком, що розв'язує, а $i$-й стовпець - стовпцем, що розв'язує.

Подальший новий вибір базисних змінних має бути знайдений із рівнянь (10) через небазисні змінні подібно до рівняння (26). Коефіцієнти $a_{ji}$ у цьому рівнянні є новою симплексною таблицею. Вони обчислюються так:

$$a'_{ji} = a_{ji} - a_{ri}\frac{a_{jl}}{a_{rl}}, i \neq l, \tag{51}$$

$$b'_j = a'_{j0} = a_{j0} - a_{r0}\frac{a_{jl}}{a_{rl}},\ j \neq r, \tag{52}$$

$$a'_{jl} = -\frac{a_{jl}}{a_{rl}},\ j \neq r, \tag{53}$$

$$b'_r = a'_{r0} = \frac{a_{r0}}{a_{rl}}, \tag{54}$$

Аналогічним шляхом обчислюються величини $z_i$:

$$z'_i = z_i - z_l\frac{a_{ri}}{a_{rl}}, \quad z'_l = -\frac{z_l}{a_{rl}}. \tag{55}$$

Нове допустиме розв'язання:

$$\begin{aligned} &x_i = 0 && i \neq l, i = \overline{1, N} \\ &x_{n+r} = && 0; \\ &x_{N+j} = b_j - a_{ji}\frac{b_r}{a_{rl}},\ j \neq r,\ j = \overline{1, m}; \\ &x_{N+r} = \frac{b_r}{a_{rl}}. \end{aligned} \tag{56}$$

У цій системі рівнянь індекс $l$ відповідає індексу тієї змінної $x_i$, якій привласнено значення $l$.

## 2.2.4. Приклади

*Приклад 1*. Розглянемо приклад, який було розв'язано у графічний спосіб. Задача полягає у такому.

$$\max\{x_1 + 4x_2 \mid x_1 \le 10,\ x_2 \le 10,\ x_1 + x_2 \le 14,\ x_1, x_2 \ge 0\}. \tag{57}$$

Запровадимо нові змінні. Отримуємо

$$\begin{aligned} x_1 + x_2 + x_3 &= 14, \\ x_1 + x_4 &= 10, \\ x_2 + x_5 &= 10. \end{aligned} \qquad (58)$$

У даному випадку

$$c^T = [1\,4]^T; \quad b^T = [14\,10\,10].$$

$$A = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix}.$$

Початкові базисні змінні -- $x_3 = 14$, $x_4 = 10$, $x_5 = 10$. Обчислимо:

$$f(x) = 0, \quad z_1 = 1, z_2 = 4.$$

$$a = \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Провідний елемент -- $a_{3,2} = 1$, ( $z_2 = 4 > z_1$ ).

оскільки стовпці таблиці, що відповідають базисним змінним, утворюють одиничну матрицю, немає потреби відтворювати її на кожній ітерації. Оскільки $c_1 = c_2 = c_3 = 0$,

$$f_0(x) = \sum_{j=1}^{3} c_j x_j = 0.$$

За допомогою (31) обчислюємо

$$z_1 = c_1 = 1 > 0, \quad z - 2 = c_2 = 4 > 0.$$

Оскільки $| z_2 |= 4 >| z_1 |= 1$, для включення в базис обираємо $x_2$. Тепер

$$\frac{b_1}{a_{1,2}} = 14 > \frac{b_2}{a_{3,2}} = 10.$$

Таким чином, як провідний рядок обираємо третій рядок, тобто змінна $x_5$ виводиться з базису, а $a_{3,2}$ є провідний елемент. Згідно з формулами ((49))-((56)) обчислюється нова таблиця.

Базисні змінні $x_3 = 4$, $x_4 = 10$, $x_2 = 10$; $f(x) = 40$.

$$a = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

$$z_1 = 1; \quad z_2 = -4.$$

Провідний -- перший стовпець. Оскільки

$$\frac{b'_1}{a'_{1,1}} = 4 < \frac{b'_2}{a'_{2,1}} = 10,$$

з базису виводиться $x_3$.

На наступній ітерації отримуємо: $x_1 = 4$, $x_4 = 6$; $x - 2 = 10$; $f(x) = 44$.

$$a = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}.$$

$$x_1 = -1;\ x_2 = -4.$$

Таким чином, одержано оптимальне розв'язання

$$x_1 = 4,\ x_2 = 10,\ f_{\text{opt}} = 44.$$

*Приклад 2*.

Знайти мінімум функції

$$z = -3x_1 - 5x_2 - 8x_3$$

із обмеженнями

$$\begin{aligned} 4x_1 + x_2 + x_3 &\le 12, \\ x_1 + 2x_3 &\le 6, \\ 3x_1 &\le 9, \\ 2x_2 &\le 4, \\ 4x_3 &\le 16. \end{aligned}$$

Запровадимо базисні змінні $x_4$ -- $x_8$ (за кількістю обмежень) і запишемо систему обмежень у нормальній формі:

$$\begin{aligned} 4x_1 + x_2 + x_3 + x_4 &= 12, \\ x_1 + 2x_3 + x_5 &= 6, \\ 3x_1 + x_6 &= 9, \\ 2x_2 + x_7 &= 4, \\ 4x_3 + x_8 &= 16. \end{aligned}$$

Перше базисне розв'язання дорівнює правій частині системи лінійних рівнянь:

$$x_4 = 12,\ x_5 = 6,\ x_6 = 9,\ x_7 = 4,\ x_8 = 16, \quad x_1 = x_2 = x_3 = 0;\ z = 0.$$

Небазисні змінні входять у функцію, що мінімізується, з від'ємними коефіцієнтами, причому третій з них за модулем більший за перші два. Отже, змінну $x_3$ слід ввести в базис. Щоб визначити яку з базисних змінних слід вивести із базису, запишемо обмеження у вигляді, розв'язаному відносно базисних змінних. Маємо:

$$x_4 = 12 - x_3,\ x_5 = 6 - 2x_3,\ x_7 = 4,\ x_8 = 16 - x_3.$$

При зростанні значень $x_3$ першою перетвориться на нуль змінна $x_5 = 0$ при $x_3 = 3$. Її й виводимо із базису. Замістимо у функції мети базисну змінну $x_3$ на небазисну $x_5$: $x_3 = 96 - x_3)/2$.

$$z = -3x_1 - 5x_2 - 4(6 - x_5) = -24 - 3x_1 - 5x_2 + 4x_5;\ z_1 = -24 < z_0.$$

Новий базис:

$$x_3 = 3,\ x_4 = 9,\ x_6 = 9,\ x_7 = 4,\ x_8 = 4;\ x_1 = x_2 = x_5 = 0.$$

Оскільки тепер найменший коефіцієнт в $z$ -- при $x_2$, вводимо цю змінну в базис.

$$x_4 = 12 - x_3 - x_2 - 4x_1 = 9 - x_2,\ x_7 = 4 - 2x_2.$$

Звідси маємо $x_7 = 0$ при $x_2 = 2$. Новий базис:

$$x_2 = 2,\ x_3 = 3,\ x_4 = 7,\ x_6 = 9,\ x_8 = 4 : x_1 = x_3 = x_7 = 0.$$

Заміщуємо в функції мети базисну змінну $x_2$ її виразом $2x_2 = 4 - x_7$. Отримуємо

$$z = -24 - 3x_1 - 5(2 - x_7/2) + 4x_5 = -34 - 3x_1 + 4x_5 + 5/2x_7;\ z_2 = -34 < z_1.$$

Ця функція має від'ємний коефіцієнт тільки при $x_1$. Вводимо його у базис:

$$x_4 = 12 - x_3 - x_2 - 4x_1 = 7 - 4x_1;\ x_6 = 9 - 3x_1,\ x_8 = 16 - 4x_1.$$

Отримуємо $x_4 = 0$ при $x_1 = 7.4$. Новий базис:

$$x_1 = 7/4,\ x_2 = 2,\ x_3 = 3,\ x_6 = 15/4,\ x_8 = 9;\ x_4 = x_5 = x_7 = 0.$$

Оскільки згідно з рівнянням (43) змінна $x_1$ входить у перші три рівняння цієї системи, знайдемо вираз для $x_1$ через небазисні з мінні $x_4$, $x_5$ та $x_7$. Маємо:

$$4x_1 = 12 - x_2 - x_3 - x_4 = 12 - 2 + x_7/2 - 3 + x_5/2 + x_1/2 - x_4 \rightarrow$$

$$x_1 = 2/6 \cdot (7 - x_4 + x_5/2 + x_7/2).$$

Підставимо цей вираз у функцію мети.

$$z = -34 - 6/7(7 - x_4 + x_5/2 + x_7/2) + 4x_5 + 5/2x_7 = -32 + 6/7x_4 + 25/7x_5 + 2x_7.$$

Оскільки всі коефіцієнти при небазисних змінних додатні, досягнуто мінімум функції мети:

$$z = -40;\ x_1 = 7/4,\ x_2 = 2,\ x_3 = 3.$$

## 2.3. Канонічна форма задачі ЛП

У тому випадку, коли задача лінійного програмування задана в канонічній формі, вибір першого базису може бути здійснений за допомогою запровадження штучних змінних $\xi_i$, що дорівнюють за умовою нулю.

Нехай задано задачу ЛП у канонічній формі

$$\begin{aligned} &E = c_0 + \sum_{j=1}^{n} c_j x_j \quad \rightarrow \min(\max); \\ &\sum_{j=1}^{n} a_{ij} x_j = b_i, \qquad x_j \geq 0,\ j = \overline{1, m}. \end{aligned} \tag{59}$$

Запишемо обмеження у стандартній формі таким чином:

$$\xi_i = b_i - \left( \sum_{j=1}^{n} a_{ij} x_j \right) = 0,\ i = \overline{1, m}. \tag{60}$$

За такої форми запису в перший базис входять $m$ штучних змінних $\xi_i$, що дорівнюють за умовою нулю. Після складання першої стандартної таблиці слід всі штучні змінні вивести із базису, а на їхнє місце ввести основні змінні, що були вільними ($x_j$). Після переводу штучної змінної у вільну весь стовпець, що відповідає цій змінній, слід викреслити. Таким чином, після виводу всіх штучних змінних $\xi_i$ у вільні та їх виключення всі основні змінні виявляться розділеними на базисні та вільні, після чого розв'язання задачі виконується звичайним чином.

Ця задача еквівалентна розв'язанню системи лінійних алгебраїчних рівнянь із $n$

змінними та $m < n$ рівняннями, розв'язання якої відомо із курсу лінійної алгебри, де $n-m$ змінних вважаються вільними і входять у систему лінійних рівнянь як вільні параметри, загальне ж розв'язання відшукується відносно основних $m$ змінних. Ці (основні) змінні визначаються шляхом визначення мінорів, що не дорівнюють нулю.

Табл.2.1. Стандартна таблиця

| | Вільний член | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ |
|---|---|---|---|---|---|---|
| $e$ | 11 | 1 | 2 | 2 | 4 | -3 |
| $E$ | 6 | 2 | -1 | 3 | -2 | -10 |
| $\xi_1$ | 2 | 1 | -1 | 2 | -2 | -6 |
| $\xi_2$ | 5 | 1 | 2 | -1 | 7 | 3 |
| $\xi_3$ | 4 | -1 | 1 | 1 | -1 | 0 |

Такий підхід до вирішення проблеми цілком слушний, але при цьому не враховується функція мети, пошук екстремуму якої й є головною метою розв'язання задачі лінійного програмування.

*Приклад.*

Дана задача ЛП:

$$\begin{aligned} E &= 6-(2x_1 - x_2 + 3x_3 - 2x_4 - 10x_5) \quad \to \min; \\ x_1 - x_2 + 2x_3 - 2x_4 - 6x_5 &= 2, \\ x_1 + 2x_2 - x_3 + 7x_4 + 3x_5 &= 5, \\ -x_1 + x_2 + x_3 - x_4 + 0 \cdot x_5 &= 4, \\ x_j \ge 0,\ j &= \overline{1,5}. \end{aligned} \tag{61}$$

Запровадимо штучні змінні і запишемо умови задачі в стандартній формі:

$$\begin{aligned} E = \quad & 6-(2x_1 - x_2 + 3x_3 - 2x_4 - 10x_5) \quad \to \min; \\ \xi_1 = \quad & 2-(x_1 + x_2 - 2x_3 - 2x_4 - 6x_5), \\ \xi_2 = \quad & 5-(x_1 + 2x_2 - x_3 + 7x_4 + 3x_5), \\ \xi_3 = \quad & 4-(-x_1 + x_2 + x_3 - x_4). \end{aligned} \tag{62}$$

Додатково запровадимо допоміжну функцію мети

$$e = \xi_1 + \xi_2 + \xi_3. \tag{63}$$

і замістимо у ній значення $\xi_i$ із останньої системи рівнянь. Отримаємо

$$e = 11-(x_1 + 2x_2 + 2x_3 + 4x_4 - 3x_5). \tag{64}$$

Запишемо у верхніх лівих кутах таблиці значення коефіцієнтів у стандартній формі. Після цього слід почати обмін змінних. Із базисних змінних треба вивести всі штучні змінні $\xi_1$, $\xi_2$, $\xi_3$, а на їхнє місце ввести основні. Питання полягає у тому, як вибирати при цьому стовпець, що вирішує.

При виборі стовпця, що вирішує, будемо виходити із задачі мінімізації допоміжної цільової функції $e$. У цьому випадку отримання мінімального значення $e = 0$ буде відповідати опорному розв'язанню задачі. Під час розв'язання за цим варіантом попередньо

треба привести всі вільні члени до вигляду $b_i \geq 0$ шляхом множення лівої та правої частин обмежень, що містять $b_i < 0$, на -1. Після отримання опорного розв'язання слід перейти до відшукання оптимального розв'язання.

Табл.2.2. Таблиця після $\xi_1 \leftrightarrow x_1$

| | Вільний член | $x_2$ | $x_3$ | $x_4$ | $x_5$ |
|---|---|---|---|---|---|
| $e$ | 9 | 3 | 0 | 6 | 3 |
| $E$ | 2 | 1 | -1 | 2 | 2 |
| $x_1$ | 2 | -1 | 2 | -2 | -6 |
| $\xi_2$ | 3 | 3 | -3 | 9 | 9 |
| $\xi_3$ | 6 | 0 | 3 | -3 | -6 |

Отже, починаємо з обміну $\xi_1 \leftrightarrow x_1$ . Маємо із першого обмеження: $2 = x_1 - x_2 + 2x_3 - 2x_4 - 6x_5 \rightarrow x_1 = 2 + x_2 - 2x_3 + 2x_4 + 6x_5$ . Функція мети

$$e = 11 - (2 + x_2 - 2x_3 + 2x_4 + 6x_5 + 2x_2 + 2x_3 + 4x_4 - 3x_5) = 9 = 3x_2 + 0x_3 - 6x_4 + 3x_5.$$

Решта обмежень: $\xi_2 = 3 - (3x_2 - 3x_3 + 9x_4 + 9x_5)$ ; $\xi_3 = 6 - (3x_3 - 3x_4 - 6x_5)$ ;

$$E = 2 - (x_2 - x_3 + 2x_4 + 3x_5).$$

Виводимо із базису $\xi_2$ : $\xi_2 \leftrightarrow x_2$ . $3x_2 = 3 - (-3x_3 + 9x_4 + 9x_5)$ ; $e = 9 - (3 + 3x_3 + 9x_4 + 9x_5) = 6 - (3x_3 - 3x_4 - 6x_5)$ ; $\xi_3 = 6 - (3x_3 - 3x_4 - 6x_5)$ .

$$E = 1 - (-x_4 - x_5).$$

Табл.2.3. Таблиця після $\xi_2 \leftrightarrow x_2$

| | Вільний член | $x_3$ | $x_4$ | $x_5$ |
|---|---|---|---|---|
| $e$ | 6 | 3 | -3 | 6 |
| $E$ | 1 | 0 | -1 | -1 |
| $x_1$ | 3 | 1 | 1 | -3 |
| $x_2$ | 1 | -1 | 3 | 3 |
| $\xi_3$ | 6 | 3 | -3 | -6 |

Після обміну $\xi_3 \leftrightarrow x_3$ за тим самим правилом отримуємо

Табл.2.4. Опорне розв'язання

| | Вільний член | $x_4$ | $x_5$ |
|---|---|---|---|
| $e$ | 0 | 0 | 0 |
| $E$ | 1 | -1 | -1 |
| $x_1$ | 1 | 2 | 1 |

| | | | |
|---|---|---|---|
| $x_2$ | 3 | 2 | 1 |
| $x_3$ | -2 | -1 | -2 |

У цій таблиці допоміжна функція мети дорівнює нулю, що відповідає опорному розв'язанню. Дійсно, вільні члени для всіх змінних додатні, що відповідає опорному розв'язанню. Зауважимо, що якщо виявиться, що допоміжна функція мети виявиться додатною, то це означає, що обмеження несумісні і ОДР не існує.

Після цього переходимо до визначення оптимального розв'язання. У рядку для функції мети $E$ коефіцієнти біля $x_4$ та $x_5$ від'ємні, отже отримано мінімальне значення функції $E$, що дорівнює $\min E^* = 1$ зі значеннями змінних: $x_1^* = 1$, $x_2^* = 3$, $x_3^* = 2$, $x_4^* = 0$, $x_5^* = 0$.

Зазначимо, що наведений метод надає можливість знаходити та виключати лінійно-залежні рівняння. Якщо початкова задача містить лінійно-залежні рівняння, у процесі виведення штучних змінних коефіцієнти рядку таких рівнянь будуть дорівнювати нулю. У прикладі, що розглянуто, лінійно-залежним рівнянням була допоміжна функція мети $e$.

Якщо задача ЛП задана у змішаній формі, тобто присутні обмеження як у вигляді рівностей, так й у вигляді нерівностей, перший базис буде включати додаткові змінні $x_{n+i}$ ( $i = \overline{1, p}$ ) для нерівностей та штучні змінні $\xi_i$ ( $i = \overline{p+1, m}$ ) для рівностей. Після формування першого базису слід спершу виключити із базису штучні змінні, а потім відшукати опорне розв'язання та оптимальне розв'язання у звичайному порядку.

## 2.4. Транспортні задачі і методи їхнього розв'язання

Постановку транспортної задачі розглянуто у вступі.

Нехай маємо m постачальників деякого виду продукції. Максимальні обсяги можливих постачань відомі й дорівнюють $a_i$, $i = \overline{1, m}$. Ця продукція використовується п споживачами. Обсяги споживання також відомі й дорівнюють $\theta_j$, $j = \overline{1, n}$. Вартість перевезення одиниці продукції від $i$-го постачальника до $j$-го споживача відома для всіх $i = \overline{1, m}$ і дорівнює $c_{ij}$. Необхідно визначити такі обсяги перевезень $x_{ij}$, щоб сумарні витрати на перевезення були мінімальними і задовольнялись вимоги всіх споживачів (за умови, що загальний обсяг можливих поставок покриває загальний обсяг заказів).

Математична модель цієї задачі така:

$$\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij} x_{ij} \to \min; \tag{65}$$

$$\sum_{j=1}^{n} x_{ij} \le a_i, \; i = \overline{1, m} \tag{66}$$

$$\sum_{i=1}^{m} x_{ij} \ge \theta_i, \; x_{ij} \ge 0, \; j = \overline{1, n}. \tag{67}$$

Вочевидь, це -- задача лінійного програмування з $m \cdot n$ змінними та $m + n$ непрямими обмеженнями. Розглянемо спочатку випадок, коли сума можливих поставок точно дорівнює сумі заказів (потреб).

$$\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} \theta_j = d.$$

Це -- так звана закрита транспортна задача. У цьому разі модель має канонічний вигляд

$$\begin{aligned} &\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} \quad \to \min; \\ &\sum_{j=1}^{n} x_{ij} = \quad a_i,\ i = \overline{1,m} \\ &\sum_{i=1}^{n} x_{ij} = \quad \theta_j,\ j = \overline{1,n}, \\ &x_{ij} \geq \quad 0. \end{aligned} \tag{68}$$

Визначимо основні властивості задачі (53).

1) Задача (53) у будь-якому випадку допустима й вирішувана. Дійсно, приміром, набір значень невідомих

$$x_{ij} = \frac{a_i\theta_j}{d},\ i = \overline{1,m},\ j = \overline{1,n}$$

задовольняє всім обмеженням задачі: вони невід'ємні й

$$\sum_{j=1}^{n} \frac{a_i\theta_j}{d} = \frac{a_i}{d}\sum_{j=1}^{n}\theta_j = a_i,\ i = \overline{1,m};$$

$$\sum_{i=1}^{n} \frac{a_i\theta_j}{d} = \frac{\theta_j}{d}\sum_{i=1}^{n} a_i = \theta_j,\ j = \overline{1,n}.$$

Вирішуваність задачі витікає з обмеженості допустимого многогранника

$$x_{ij} \leq \min(a_i, \theta_j).$$

Табл.2.5. Опорне розв'язання

| 1 | | 2 | | | n | | $a$ |
|---|---|---|---|---|---|---|---|
| $c_{11}$ | | $c_{12}$ | | | $c_{1n}$ | | $a_1$ |
| | $x_{11}$ | | $x_{12}$ | | | $x_{1n}$ | |
| $c_{21}$ | | $c_{22}$ | | | $c_{2n}$ | | $a_2$ |
| | $x_{21}$ | | $x_{22}$ | | | $x_{2n}$ | |
| | | | | | | | |
| $c_{m1}$ | | $c_{m2}$ | | | $c_{mn}$ | | $a_m$ |
| | $x_{m1}$ | | $x_{m2}$ | | | $x_{mn}$ | |
| $\theta_1$ | | $\theta_2$ | | | $\theta_n$ | | $d$ |

2) Ранг матриці системи рівнянь-обмежень задачі (53) дорівнює .

3) Якщо в задачі ((68)) всі $a_i$ та $\theta_j$ -- цілі числа, бодай одне її оптимальне розв'язання є цілочислове. Шляхом вибору відповідних одиниць виміру продукції завжди можна від нецілих значень $\theta_j$ та $a_i$ перейти до цілих. Таким чином, умова цілочисельності $\theta_j$ та $a_i$ не

обмежує узагальненості, тож можна вважати, що задача (53) має цілочислове оптимальне розв'язання.

Запишемо задачу у вигляді табл. 1.5.

Кожна клітинка таблиці (за винятком правого стовпця та нижнього рядку) відповідає конкретній парі постачальник - споживач. У комірки будемо заносити обсяги перевезень за відповідним маршрутом.

Будемо будувати допустиме розв'язання, починаючи із встановлення обсягів перевезень за маршрутом (1,1), тобто заповнення верхньої лівої (``північно - західної'') комірки таблиці. Нехай він буде максимально можливим за умовами задачі, тобто дорівнює

$$\overline{x}_{11} = \min(a_1, \theta_1).$$

Якщо $a_1 < \theta_1$, постачальник 1 сповна використав свої можливості, отже при визначенні решти перевезень його можна не враховувати, потреба першого споживача тепер буде дорівнювати $\theta_1 - a_1$, якщо $\theta_1 < a_1$, перший споживач сповна задовольнив свої потреби у продукції й його можна далі не враховувати, а постачальний 1 тепер має у своєму розпорядженні лише $a_1 - \theta_1$ одиниць продукції. Якщо $a_1 = \theta_1$? далі можна не розглядати ані постачальника, ані споживача. Але умовимось вважати, що у цьому випадку ``з гри вибув'' тільки один з них (нехай, для визначеності, постачальник), а можливі поставки (або, відповідно, потреби) іншого дорівнюють нулю.

Табл.2.6. Допустимий розв'язок задачі (53)

| 1 | | 2 | | 3 | 4 | 5 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0,7 | - | 0.5 | | 0,6 | 0,9 | 0,5 | | | | |
| | 100 - | --- | $\gamma$ | | | | 100 | | 0 | |
| 0,6 | ⋮+ | 0,5 | ⋮- | 0,8 | | 0,8 | 1,0 | | | |
| | 20- | --- | 130 | | | | 150 | 130 | 0 | |
| 0,3 | | 0,2 | | 0,5 | 0,4 | 0,4 | | | | |
| | | 70 | | 100 | 30 | | 200 | 130 | 30 | 0 |
| 0,9 | | 1,1 | | 1,0 | 0,8 | 1,1 | | | | |
| | | | | | 0 | 100 | 100 | 100 | 0 | |
| 120 | | 200 | | 100 | 30 | 100 | 550 | | | |
| 20 | | 70 | | 0 | 0 | 0 | | | | |
| 0 | | 0 | | | | | | | | |

Після встановлення обсягів перевезень за маршрутом (1,1) маємо нову задачу вигляду (53), в якій сумарне число постачальників та споживачів на одиницю менше за початкову. У північно-західну комірку таблиці цієї нової задачі знову розміщуємо максимально можливий обсяг перевезень (може статися, що він дорівнює нулю). Продовжуючи цей процес, прийдемо до допустимого розв'язання задачі, оскільки

$$\sum_{i=1}^{m} a_i = \sum_{j=1}^{m} \theta_j = d.$$

Засіб побудови допустимого розв'язання задачі (53), що викладено, має назву методу північно-західного кута.

*Приклад*. Знайдемо допустиме розв'язання задачі (53), що визначається табл. 2.6. Ця таблиця містить перевезення, що отримано за методом північно-західного кута, а також вказано послідовно обсяги можливих поставок та заказів. За цими вказівками неважко відновити процес послідовного заповнення північно-західних клітинок. Отримано таке допустиме розв'язання.

$$x_{11} = 100,\ x_{21} = 20,\ x_{22} = 130,\ x_{33} = 100,\ x_{34} = 30,\ x_{45} = 100.$$

Решта змінних $x_{ij} = 0$.

Метод північно-західного кута завжди призводить до опорного допустимого розв'язання, отже далі для пошуку оптимального розв'язання можна застосовувати симплекс-метод.

При побудові початкового опорного розв'язання за методом північно- західного кута не враховуються питомі витрати $c_{ij}$ на перевезення продукції. Між тим вірогідно, що чим більше у початковому опорному розв'язанні дешевих перевезень, тим ближче це розв'язання до оптимального. Вибір саме північно-західного кута в кожній з послідовних часткових таблиць - лише одне з можливих правил вибору одного із маршрутів, що розглядаються. Принципово нічого не зміниться, якщо обирати у початковій таблиці, а потім у її частинах, що послідовно розглядаються, наприклад, комірку, яка відповідає самому дешевому маршруту, й вводити, як і раніше, максимально можливе перевезення. Така побудова вихідного опорного розв'язання називається побудовою за правилом *мінімального елементу*. Можливі й інші підходи до побудови правил вибору маршруту (за першим рядком, першим стовпцем, в яких вибирається

мінімальний елемент $c_{ij}$ тощо).

### 2.4.1. Метод потенціалів

Розглянемо застосування симплекс-методу до розв'язання транспортної задачі (метод потенціалів).

Перш за все врахуємо, що транспортна задача -- це задача на мінімум, тобто у симплекс-алгоритмі від'ємні та додатні оцінки $\Delta_i$ міняються місцями. В умовах транспортної задачі більш зручно співставляти обмеженням задачі змінні із окремими позначеннями для першої (постачальниками) та другої (споживачами) групи обмежень. А саме, першому обмеженню, що пов'язане із першим постачальником, поставимо у відповідність змінну $u_1,\ldots,u_m$, першому обмеженню другої групи $v_1,\ldots,v_n$. Ці змінні складають вектор

$$w=[u_1,u_2,\ldots,u_m,v_1,v_2,\ldots,v_n].$$

Нехай задано якесь опорне розв'язання транспортної задачі та його базис. Складемо систему рівнянь відносно у якій коефіцієнти при невідомих співпадають із координатами векторів $A_{ij}$, що відповідають базисним векторам $A'_{ij}$. Оскільки вектор

$$A_{ij}=00,\ldots,1_{(i)},\ldots,1_{(m+j)},\ldots,0]^T,$$

то відповідне до нього рівняння набуває вигляду

$$u_i+v_j=c_{ij}. \qquad (69)$$

У цій системі рівнянь $(i, j)$ пробігають *усі базисні* пари індексів.

Оскільки в системі обмежень транспортної задачі одне з обмежень зайве, можна прирівняти відповідну до нього змінну нулю. Оскільки зайвою можна вважати будь-яку змінну, значення 0 можна привласнити буд-якій зі змінних $u_i$ або $v_j$. Нехай $u_i=0$. Тоді система рівнянь (54) перетворюється у систему з $n+m-1$ рівнянь із $n+m-1$ невідомими, в якій вектори-рядки коефіцієнтів збігаються з базисними, й отже лінійно незалежними векторами $A'_{ij}$, де (i, j) - базисні пари індексів.

Тепер знайдемо оцінки $\Delta_i$ для всіх небазисних векторів (для базисних вони дорівнюють нулю).

$$\Delta_i=u_i+v_j-c_{ij}. \qquad (70)$$

Якщо всі $\Delta_i<0$, отримане опорне розв'язання є оптимальним. Якщо серед оцінок є додатні, треба переходити до нового опорного розв'язання.

Специфіка транспортної задачі дозволяє відшукувати нове опорне розв'язання та новий базис за спрощеним порівняно з загальним випадком правилом. Нехай $\Delta_{i_0j_0}$ -- одна із додатних оцінок (приміром, найбільша, або перша додатна у ході її обчислення). У цьому випадку в базис вводиться вектор із додатних оцінок $A_{i_0j_0}$. Позначимо обсяг перевезень за маршрутом ( $i_0j_0$ ) у новому базисному розв'язанні через $\gamma$. Оскільки сумарний обсяг перевезень із $i$-го пункту у допустимому розв'язанні має дорівнювати $a_i$, перевезення обсягу $\gamma$ за маршрутом ( $i_0j_0$ ) має компенсуватися зменшенням на $\gamma$ сумарного обсягу старих перевезень у рядку $i_0$.

Нехай, наприклад, ми вирішили зменшити перевезення за маршрутом ( $i_0 j_1$ ) за рахунок збільшення одного (базисного) перевезення у стовпці $j_1$. Далі, нехай ми вирішили збільшити на $\gamma$ перевезення за маршрутом ( $i_0 j_1$) . Тоді необхідно компенсувати це збільшення за рахунок зміншення на $\gamma$ обсягу одного базисного перевезення у рядку $i_1$ . Якщо продовжувати такі послідовні зміни обсягів перевезення, які відновлюють баланс за рядками та стовпцями, дійдемо до необхідності компенсувати введення перевезення обсягу $\gamma$ за рахунок зменшення на $\gamma$ базисного перевезення у стовпці $j_0$ . Ланцюжок комірок $(i_0, j_0),(i_0, j_1),(i_1, j_1),(i_1, j_2),\ldots$ має замкнутись на комірці ( $i_0, j_0$ ), причому такий ланцюжок може бути тільки один.

Звідси витікає, що за належного значення $\gamma > 0$ в одній із комірок отримаємо нульове перевезення й ця комірка виключається із базису, а решта комірок разом із старими базисними комірками утворюють новий базисний набір перевезень, що не дорівнюють нулю.

Значення $\gamma$ визначається просто: $\gamma$ *дорівнює мінімальному базисному перевезенню*, що є у старому ланцюжку, із яких віднімається $\gamma$ . Вочевидь, одне із старих перевезень перетворюється на нуль. Відповідний вектор виводиться з базису.

Розглянемо *приклад*. Нехай треба перейти до нового опорного розв'язання в задачі, що задана табл. 2.7. Старе опорне розв'язання отримано за методом північно-західного кута. Для визначення оптимальності цього опорного розв'язання і вектору, який слід ввести у базис у разі, коли розв'язання не є оптимальним, знайдемо оцінки небазисних векторів. З цією метою складемо систему рівнянь (53). У нас базисні пари індексів (1,1) (2,1) , (2, 2), (3, 2), (3, 3), (3, 4), (4,4), (4, 5). Відповідні значення вартостей

$$c = [0,7\ 0,4\ 0,5\ 0,2\ 0,5\ 0,4\ 0,8\ 1,1].$$

Система рівнянь набуває вигляду

$$u_1 + v_1 = 0,7,\ u_2 + v_1 = 0,4,\ u_2 + v_2 = 0,5,\ u_3 + v_2 = 0,2;$$

$$u_3 + v_3 = 0,5;\ u_3 + v_4 = 0,4;\ u_4 + v_4 = 0,8;\ u_4 + v_5 = 1,1.$$

Нехай $u_1 = 0$, послідовно отримуємо

$$v_1 = 0,7;\ u_2 = -0,3;\ v_3 = 0,8;\ u_3 = -0,6;\ v_3 = 1,1;\ v_4 = 1,0;\ u_4 = -0,2;\ v_5 = 1,3.$$

Обчислюємо

$$\Delta_{1,2} = u_1 + v_2 - c_{1,2} = 0 + 0,8 - 0,3 = 0,5.$$

Тобто перша оцінка, що обчислена, виявилась додатною. Значить, в базис треба вводити вектор $A_{1,2}$ . У комірку (1, 2) занесемо $\gamma \geq 0$ . В табл. (185) позначка ``—" означає, що із відповідного перевезення $\gamma$ має відніматися, а ``+" - що $\gamma$ має додаватися до цього перевезення.

Згідно з правилом знаходження $\gamma$ , значення $\gamma$ дорівнює найменшому з усіх перевезень ланцюжка, із якими пов'язана позначка —. Тобто $\gamma = 100$ . Перевезення за маршрутом (1,1) з $\gamma$ на нульове. Новий ланцюжок розв'язання наведено у табл. 2.7.

Табл.2.7. Опорне розв'язання задачі

| 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|
| | | | | |
| | 100 -- | -- $\gamma$ | | |
| | | | | |
| 100 | 30 | | | |
| | + | - | | |
| | 70 -- | -- 100 | 30 | |
| | | | 0 | 100 |

Із цієї таблиці видно, що у разі розв'язання відносно невеликих задач можна відшукати компенсуючий ланцюжок та визначити вектор, який має виводитися із базису та значення $\gamma$, безпосередньо із таблиці. Але для великих задач ця процедура може виявитися досить непростою.

Викладений алгоритм одержав назву методу потенціалів, оскільки у деякому сенсі суми $u_i + v_i$ можна розглядати як аналоги кількості роботи, що необхідна для переміщення одиничної маси з пункту $i$ у пункт $j$ у потенційному полі. Величини $u_i$, $v_i$ прийнято називати потенціалами витоків продукції та пунктів споживання відповідно.

Алгоритм методу потенціалів записується так.

Нехай дано транспортну задачу. Для її розв'язання необхідно:

1. Знайти початкове опорне розв'язання (наприклад, за методом північно-західного кута або за методом мінімального елементу).

2. Знайти значення потенціалів $u_i$, $v_j$ шляхом розв'язання системи рівнянь (54).

3. Для всіх небазисних пар індексів $(i, j)$ обчислити оцінки $\Delta_{ij}$ за формулою (55). Якщо всі оцінки $\Delta_{ij} < 0$, процес завершується: опорне розв'язання оптимальне. Якщо серед $\Delta_{ij}$ є додатні, то за заздалегідь встановленим правилом вибирається одна із цих оцінок (приміром, максимальна, або перша оцінка, що виявилась додатною).

4. Користуючись правилом переходу до нового опорного розв'язання у тій чи іншій формі (наприклад, табличній), побудувати нове опорне розв'язання шляхом введення у число базисних пар індексів пару що вибрано, й вивести із числа базисних пар одну зі старих базисних пар.

Табл. 2.8. Опорне розв'язання задачі

| 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|
| | | - | | |
| | | 100 -- | -- -- | -- 0 |
| 120 | 30 | | | |
| | | + | -- | |
| | 170 | 0-- | --30 | |
| | | | + | - |
| | | | 0-- | --100 |

Повернутись до п. 2, маючи на увазі нове опорне розв'язання.

Гарантується, що після обмеженого числа циклів (ітерацій), описаних вище, буде отримано оптимальне опорне розв'язання, тобто оптимальний план перевезень продукції від постачальника до споживача.

Повернемось до прикладу. Початкове опорне розв'язання наведено у табл. 2.6. Перехід до нового опорного розв'язання наведено у табл. 2.7 Обчислимо відповідні до нього потенціали $u_i$ та $v_i$ й оцінки $\Delta_{ij}$. Звідси отримуємо:

$$u_1 = 0; u_2 = 0; u_3 = -0,3; u_4 = 0,1; v_1 = 0,4; v_2 = 0,5; v_3 = 0,8; v_4 = 0,7; v_5 = 1,0$$

$$\Delta_{1,1} = u_1 + v_1 - c_{1,1} = -0,3; \Delta_{1,2} = u_1 + v_2 - c_{1,2} = 0; \Delta_{1,3} = u_1 + v_3 - c_{1,3} = 0,3 > 0.$$

Отже, треба переходити до нового опорного розв'язання. На цьому перша ітерація завершується.

Друга ітерація. Вводимо у число базисних пар (1, 3) і будуємо компенсуючий ланцюжок (табл. 2.8). Визначаємо $\gamma = 100$. За цього значення $\gamma$ нулю дорівнюють два перевезення: (1,2) та (3,3). З числа базисних перевезень виводимо (1,2). У комірку (3,3) заносимо 0 (у комірки, що відповідають небазисним змінним, нулі не заносяться). Нове опорне розв'язання представимо у вигляді табл. 2.9..

Табл. 2.9. Опорне розв'язання задачі

| 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|
| | | 70 | | 30 |
| 120 | 30 | | | |
| | 170 | 30 | | |
| | | | 30 | 70 |

Цьому розв'язанню відповідають потенціали

$$u_1 = 0; u_2 = 0,2; u_3 = -0,1; u_4 = 0,3;$$

$$v_1 = 0,2; v_2 = 0,3; v_3 = 0,5; v_4 = 0,5; v_5 = 0,8;$$

$$\Delta_{1,1} = -0,5; \Delta_{1,2} = -0,2; \Delta_{1,4} = -0,4; \Delta_{1,5} = 0,3 > 0.$$

Третя ітерація. Вводимо у число базисних пару (1,5), будуємо компенсуючий ланцюжок (табл. 2.8). Визначаємо $\gamma = 30$. Із числа базисних виводимо пару (3,4). Нове опорне розв'язання представимо у вигляді табл. 2.9. Для нього значення потенціалів:

$$u_1 = 0; u_2 = 0,2; u_3 = -0,1; u_4 = 0,6;$$

$$v_1 = 0,2; v_2 = 0,3; v_3 = 0,6; v_4 = 0,2; v_5 = 0,5;$$

$$\Delta_{1,1} = -0,5; \Delta_{1,2} = -0,2; \Delta_{1,4} = -0,7; \Delta_{2,3} = 0; \Delta_{2,4} = -0,8; \Delta_{2,5} = -0,3;$$

$$\Delta_{3,1} = -0,2; \Delta_{3,4} = -0,3; \Delta_{3,5} = 0; \Delta_{4,1} = -0,1; \Delta_{4,2} = -0,2; \Delta_{4,3} = 0.$$

Усі оцінки, що обчислено, недодатні, значить розв'язання, що побудовано на третій ітерації, є оптимальним. Оптимальні значення змінних:

$$x_{1,3} = 70; x_{1,5} = 30; x_{2,1}120; x_{2,2} = 30; x_{3,2} = 170; x_{3,3} = 30; x_{4,4} = 30; x_{4,5} = 70.$$

Решта змінних дорівнює нулю.

Умова збалансованості є обов'язковою при розв'язанні транспортних задач спеціальними методами. На практиці, проте, значно частіше доводиться мати справу із

задачами, в яких ця умова порушується. Такі задачі мають назву *відкритих*. Можливі два випадки порушення балансу:

1. Сума наявних запасів перевищує суму поданих заявок:

$$\sum_{i=1}^{m} a_i > \sum_{j=1}^{n} b_j \qquad (71)$$

2. Сума поданих заявок перевищує суму наявних запасів:

$$\sum_{i=1}^{m} a_i < \sum_{j=1}^{n} b_j \qquad (72)$$

У цих випадках транспортну задачу необхідно звести до закритого вигляду. Для цього у першому випадку запроваджується фіктивний параметр - фіктивна заявка

$$b_f = \sum_{i=1}^{m} a_i - \sum_{j=1}^{n} b_j. \qquad (73)$$

Вартість перевезень від усіх постачальників до фіктивного споживача дорівнює нулю, тобто $c_{i,f} = 0$, $i = \overline{1,m}$.. Значення $x_{i,f}$, яке отримується внаслідок розв'язання задачі, буде означати, що у постачальника $A_i$ залишилось без попиту $x_{i,f}$ вантажу. У другому випадку порушення балансу запровадиться фіктивний постачальник , якому привласнюється фіктивний запас

$$a_f = \sum_{i=1}^{n} b_i - \sum_{j=1}^{m} a_j. \qquad (74)$$

Вартість перевезення від фіктивного постачальника до всіх споживачів приймається рівною нулю: $c_{f,j} = 0$.

Значення $x_{f,j}$, яке отримується внаслідок розв'язання задачі, буде означати, що заявка постачальника $B_j$ залишилась незадоволеною на $x_{f,j}$ одиниць вантажу.

## 2.5. Цілочислове лінійне програмування

Проміж важливих з точки зору практики задач відшукання умовного екстремуму лінійної функції важливе місце займають задачі, до яких пред'являються вимоги цілочисельності всіх або частки змінних. Такі задачі одержали назву задач цілочислового програмування. На перший погляд здається цілком логічним спочатку розв'язати задачу лінійного програмування, а вже потім закруглити оптимальні значення $x_j^*$ до найближчих цілих чисел. Але такий підхід не завжди є правомірним, оскільки не відображає реальних ситуацій, що виникають на практиці, тим більше, що існує цілий клас задач, коли треба отримати точний результат.

Сформулюємо кілька задач, що призводять до задач цілочислового типу.

**Задача про ранець**.

Є $m$-вимірний вектор обмежених ресурсів $b = [b_1, b_2, \ldots, b_n]$ й які можна використати для перевезення різних за своїми характеристиками вантажів. Кожний $j$-й вантаж характеризується такими властивостями:

1) неділимістю, тобто для транспортування може вибиратись будь-який вантаж у кількості, що кратна одиниці:

2) корисністю $c_i$;

3) витратами $i$-го ресурсу для перевезення одиниці $j$-го вантажу $a_{ij}$, $j=\overline{1,m}$, $i=\overline{1,n}$ .Потрібно обрати такий набір вантажу для перевезення, за яким максимізується загальна корисність рейсу. При цьому корисність рейсу визначатимемо як сумарну вартість вантажів, перевезених за рейс.

Позначимо $x_j$ -- кількість предметів, що вибрано для транспортування, і запишемо математичну модель цієї задачі. Вочевидь, вимозі неділимості відповідає умова мо математичну модель цієї задачі. Вочевидь, вимозі неділимості відповідає умова

$$x_j \ge 0,\ x_j - \text{ö}^3\text{ë}^3,\ j=\overline{1,n} \tag{75}$$

Співставлення витрат ресурсу кожного типу для транспортування одиниці вантажу та наявності ресурсів дає обмеження

$$\sum_{j=1}^{n} a_{ij}x_j \le b_i,\ j=\overline{1,m} \tag{76}$$

Загальну корисність рейсу визначає значення функції

$$F(x)=\sum_{j=1}^{n} c_j x_j. \tag{77}$$

Частинним випадком задачі є задача про ранець, у якій будь-який із заданого набору предметів може бути або вибраний, або ні, тобто для кожного з $x_j$ допустимими є 1 або 0 (предмет вибирається або ні). Це призводить до того, що умова $x_j \ge 0$ замінюється вимогою

$$x_j = \begin{cases} 0, \\ 1 \quad , \ j=\overline{1,n}. \end{cases} \tag{78}$$

Відомо, що дискретна величина, яка набуває тільки значення 0 або 1, зветься булевою. У зв'язку з цим задача, в якій на змінні накладаються умови типу (61), отримала назву задачі із *булевими змінними*.

**Задача про вибір типу суден** (транспортних засобів). Річкове пароплавство, що здійснює обслуговування пасажирів за різними маршрутами, визначило, що у середньому за будь-яким із цих маршрутів упродовж сезону здійснюється перевезення сталої кількості пасажирів. Ефективність використання транспортних засобів пароплавства складається із ефективності обслуговування кожного з маршрутів й визначається різницею між прибутками від рейсів та витратами пароплавства на обслуговування рейсів.

Виникає задача про вибір для кожного маршруту суден таких типів та у такій кількості, що дозволили б задовольнити потреби у кількості місць для пасажирів й максимізувати прибуток пароплавства.

Нехай за $j$-м маршрутом упродовж сезону переїжджає $b_j^{(1)}$ людей. Перевезення пасажирів за цим маршрутом можна здійснити за допомогою $m$ різних типів суден. Для кожного $i$-го типу суден відомі такі характеристики:

1) $a_{i1}$ -- вантажопідйомність (кількість місць):

2) $a_{i2}$ -- кількість людей, які обслуговують перевезення:

3) $a_{i3}$ -- витрати пального упродовж сезону:

4) $c_{ij}$ -- прибуток за сезон від використання $i$-го транспортного засобу за $j$-м

маршрутом.

Треба вибрати парк суден для кожного маршруту такий, щоб максимізувати прибуток пароплавства з додержанням обмежень: загальні витрати пального за сезон не перевищують $b_3$ одиниць, загальна чисельність обслуговуючого персоналу не перевищує $b_2$ осіб.

Запишемо математичну модель сформульованої задачі. Позначимо через $x_{ij}$ -- шукану кількість суден $i$-го типу, яку доцільно додержувати для обслуговування $j$-го маршруту. Вочевидь, область пошуку розв'язання визначиться умовами:

$$\sum_{i=1}^{n} a_{i1} x_{ij} \geq b_i^{(1)},\ j = \overline{1,n}; \tag{79}$$

$$\sum_{j=1}^{n}\sum_{i=1}^{m} a_{i2} x_{ij} \leq b_2; \tag{80}$$

$$\sum_{j=1}^{n}\sum_{i=1}^{m} a_{i3} x_{ij} \leq b_3; \tag{81}$$

де $x_{ij} \geq 0$ -- цілі. У цій області потрібно знайти такі значення змінних $x = (x_{11}, x_{12}, \ldots, x_{1n}, x_{21}, x_{2,2}, \ldots, x_{2,n}, \ldots, x_{m,1}, x_{m,2}, \ldots, x_{m,n})$, які максимізують функцію

$$F(x) = \sum_{j=1}^{n}\sum_{i=1}^{m} c_{ij} x_{ij}. \tag{82}$$

У розглянутих задачах вимога цілочисельності є необхідною властивістю розв'язання, вона природно виникає в задачах: розміщення виробництва, про закупівлю парку машин для виконання виробничої програми, визначення кількості працівників на деякий термін тощо.

### 2.5.1. Постановка задачі

Проміж сукупності з $n$ неділимих предметів, кожний $j$-й з яких має по $i$-й характеристиці із показником $a_{ij}$ та корисністю $c_i$, знайти такий набір, що дозволяє максимізувати ефективність використання ресурсів величиною $b_i$ ($i = \overline{1,m}$).

Математична модель цієї задачі може бути подана таким чином:
в області, що визначається умовами

$$\sum_{j=1}^{n} a_{ij} x_j \ \{\leq = \geq\} b_i,\ i = \overline{1,m}; \tag{83}$$

$$x_j \geq 0,\ j = \overline{1,n}; \tag{84}$$

$$x_j \ --\text{ö}^3\ddot{e}^3,\ j = \overline{1,n_1}; \tag{85}$$

знайти розв'язок $x = (x_1, x_2, \ldots, x_n)$ такий, що максимізує (мінімізує) значення функції мети

$$F(x) = \sum_{j=1}^{n} c_j x_j. \tag{86}$$

Якщо $n_1 = n$ (68)--(71) є модель задачі цілочислового програмування, якщо ж $n_1 < n$ -- модель задачі частково цілочислового програмування.

До класу задач цілочислового програмування приєднуються задачі, в яких умова

цілочисельності всіх або частки змінних замінюється на вимогу дискретності.

Для кожної $j$ -ї змінної $x_j$ заздалегідь визначений набір значень (не обов'язково цілих)), які вона може приймати:

$$x_j \in A_j,\ A_j = \{0, A_{i2}, \ldots, A_{jk_j}\}.$$

Математична модель загальної задачі дискретного програмування може бути записана у такому вигляді.

$$\sum_{j=1}^{n} a_{ij} x_j \ \{\leq = \geq\} b_i,\ i = \overline{1,m}; \tag{87}$$

$$x_j \in A_j,\ A_j = \{0, A_{i2}, \ldots, A_{jk_j}\},\ j = \overline{1,n_1},\ n_1 < n. \tag{88}$$

отримати розв'язок $x = (x_1, x_2, \ldots, x_n)$ , для якого максимізується (мінімізується) лінійна функція

$$F(x) = \sum_{j=1}^{n} c_j x_j. \tag{89}$$

Для з'ясування сутності проблеми розглянемо такий простий приклад.

В області, що визначається умовами

$$\begin{aligned} 2x_1 + 11x_2 &\leq 38, \\ x_1 + x_2 &\leq 7, \\ 4x_1 - 5x_2 &\leq 5, \\ x_1 \geq 0, &\quad x_2 \geq 0, \end{aligned}$$

де $x_1, x_2$ -- цілі.

знайти максимум функції $F(x_1, x_2) = x_1 + x_2$.

Розв'яжемо задачу у геометричний спосіб.

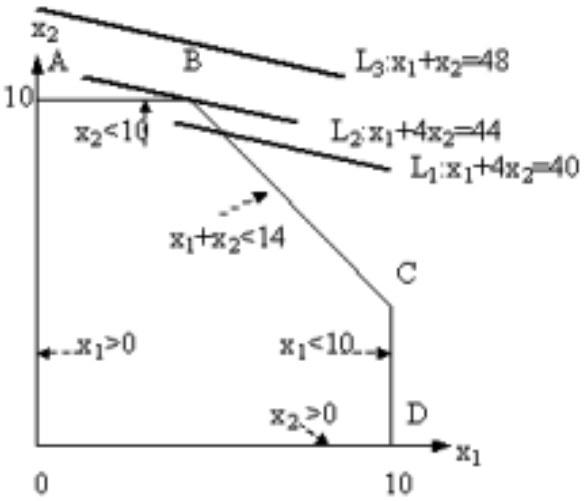


Рис. 2.3. Приклад розв’язання.

Область пошуку екстремуму - многокутник ODABC, але оскільки лінія рівню

паралельна прямій $x_1 + x_2 = 7$, екстремум функції досягається у вершинах $A(8/3,13/3)$ та $B(23/9, 40/9)$, а також у довільній точці відрізку АВ і дорівнює 7. Але нас цікавлять лише точки із цілочисловими координатами, отже, точки А, В відрізку АВ не є допустимим розв'язанням задачі. Якщо округлити координати точки А: $A'(2;4)$, то виявиться, що ця точка не належить області пошуку. Отже, для розв'язання цілочислової задачі необхідно приймати додаткові засоби. Розв'язання задачі, що розглядається, досягається в точках $N(2;3)$ та $M(3;2)$, мінімум дорівнює 5. Тобто, розв'язання задачі цілочислового програмування (на відміну від задач лінійного програмування) виявляється всередині області.

## 2.5.2. Метод Гоморі

Назвемо задачу (68) --(71) скорочено $(G^n, F)$-задачею. Задачу (68), (69), (71) (задачу лінійного програмування) назвемо $(G, F)$-задачею. Має місце

**Теорема**. *Нехай $G$ - многогранник, $G^n$ - множина цілих точок, $G^n \subset G$, $R$ -- випукла лінійна оболонка множини $G^n$. Тоді*

1) $R = R^n$ -- цілочисловий многогранник (многогранник, усі вершини якого цілочислові);

2) $R^n = G^n$;

3) $R^*$ -- множина опорних розв'язань задачі $(G^n, F)$ міститься у многограннику $R^n$.

З цієї теореми витікає висновок про те, що оптимальне розв'язання задачі, областю визначення якої є випукла оболонка, що натягнута на область пошуку цілочислового розв'язання, збігається із оптимальним розв'язанням вихідної цілочислової задачі.

Ця теорема показує принципову можливість заміни розв'язання задачі типу $(G^n, F)$ деякою процедурою розв'язання допоміжної задачі типу $(G, F)$, але ж вона не дає алгоритму вирішення.

Р. Гоморі запропонував алгоритми вирішення цілочислових та частково цілочислових задач.

**Алгоритм Гоморі** складається із таких процедур:

1. Вирішується $(G, F)$-задача, що відповідає початковій $(G^n, F)$-задачі.

2. Отримане оптимальне розв'язання $(G, F)$-задачі, за умови його існування, перевіряється на цілочисловість. Якщо умови цілочисловості виконуються за всіма змінними, оптимальне розв'язання $(G, F)$-задачі є оптимальним розв'язанням $(G^n, F)$-задачі. Якщо умова цілоцисловості не виконується бодай за однією змінною, переходять до третього етапу.

3. Будується додаткове обмеження, яке має властивість, що за його допомогою відсікається частина області, в якій міститься оптимальне розв'язання $(G, F)$-задачі і не міститься жодного допустимого розв'язання $(G^n, F)$-задачі. Процес побудови додаткових обмежень та розв'язання $(G, F)$-задач, що отримуються, триває доти, доки не отримамо цілочислове розв'язання, або не пересвідчимось у тому, що задача не має розв'язань.

Властивості, що мають місце для кожного з додаткових обмежень при переході від однієї задачі до другої:

• додаткове обмеження має бути лінійним, аби залишатися в області застосування апарату лінійного програмування:

• додаткове обмеження має відсікати частину області, в якій не містяться допустимі розв'язання цілочислової $(G^n, F)$-задачі, але є отримане оптимальне розв'язання $(G, F)$, тобто обмеження має бути правильним, яке не дозволяє втратити оптимальне розв'язання початкової $(G^n, F)$-задачі.

Нехай $x(G, F)$ -- оптимальне розв'язання $(G, F)$-задачі, яке є недопустимим розв'язанням $(G^n, F)$-задачі. Нерівність

$$\sum_{j=1}^{n} \alpha_j x_j \le \beta \qquad (90)$$

визначає правильне відсічення, якщо воно задовольняє:

• умові відсічення: $x(G, F)$ не задовольняє нерівність (75)

• умові правильності: будь-яке допустиме розв'язання $(G^n, F)$-задачі задовольняє нерівність (75).

Методи, що засновані на використанні процедури побудови правильних відсічень, отримали назву *методів відсічення*

**Реалізація алгоритму Гоморі**

1. Розв'яжемо $(G^k, F)$-задачу (спочатку $k = 0$)симплекс-методом.

Нехай у базис оптимального розв'язання увійшли вектори $A_{s,1}, \ldots, A_{s,n}$. Параметри останньої (оптимальної) симплекс-таблиці позначимо як $x_{ij}$. $\Delta_{ij} = x_{ij}$. Якщо деяка координата $x_{i,0}$ оптимального розв'язання $x(G^k, F)$ неціла, переходимо до п.2.

2. Якщо у сукупності координат оптимального розв'язання $x(G^k, F)$ маємо більше однієї нецілої, вибираємо координату із найменшим номером (для визначеності). Нехай цією координатою виявилась $x_{i,0}$. Складемо додаткове лінійне обмеження

$$x_{n+k+1} = -\{x_{l,0}\} - \sum_{j \in N} (-\{x_{l,j}\} x_j,\ x_{n+k+1} \ge 0,\ k = 0,1,2,\ldots \qquad (91)$$

3. Додамо умови (76) до умов $(G^k, F)$-задачі. Отримаємо нову, $(G^{k+1}, F)$-задачу. Оскільки оптимальне розв'язання $x(G^k, F)$ визначає одну з вершин многогранника умов, воно може бути обране як початкове опорне розв'язання для заново отриманої задачі. Це ж означає, що останню симплекс-таблицю для $(G^k, F)$-задачі можна обрати як початкову для $(G^{k+1}, F)$-задачі, якщо доповнити її умовою (76). Симплекс-таблиця для $(G^{k+1}, F)$-задачі шляхом обведення $(i+1)$-ми елементами $x_{i+1,0} = -\{x_{l_0}\}$, $x_{i+1,l} = -\{x_{l_j}\}$, $j \in N_k$, $N_k$ -- небазисні змінні $(G^k, F)$-задачі.

Змінними цієї задачі будуть $x_1, x_2, \ldots, x_n, x_{n+k+1}$ Лінійна форма нової задачі виражена через небазисні змінні $(G^k, F)$-задачі.

Рядок, що використовувався раніше для обведення, викреслюється й потім для побудови розширених задач не використовується.

*Приклад 1.*

$$
\begin{aligned}
&F(x) = 4x_1 + 5x_2 + x_3 \quad \to \max;\\
&x_1 + 4x_2 \qquad\qquad\quad \le 11,\\
&3x_1 + 2x_2 \qquad\qquad \le 10,\\
&3x_1 + 3x_2 + x_3 \qquad \le 13,\\
&x_1 \ge 0,\ x_2 \ge 0, \qquad x_3 \ge 0,\ x_i - \text{цілі}, \quad i = 1,2,3.
\end{aligned}
$$

Для сформульованої $G^n, F)$-задачі запишемо відповідну $(G^0, F))$-задачу:

$$
\begin{aligned}
&F(x) = 4x_1 + 5x_2 + x_3 \quad \to \max;\\
&x_1 + 4x_2 \qquad\qquad\quad \le 11,\\
&3x_1 + 2x_2 \qquad\qquad \le 10,\\
&3x_1 + 3x_2 + x_3 \qquad \le 13,\\
&x_1 \ge 0,\ x_2 \ge 0, \qquad x_3 \ge 0,\ i = 1,2,3.
\end{aligned}
$$

Вектор $x = (9/5; 23/10; 7/10)$ є оптимальним розв'язанням $(G^0, F))$-задачі, але не є допустимим розв'язанням $(G^1, F)$-задачі.

Переходимо до третього кроку -- побудові правильного відсічення. Як змінну, за якою будуємо відсічення, обираємо $x_3$, тобто вводимо у базис $z_1 = (x_6) = -7/10 + 9/10x_4 + 3/10x_5$ замість $x_3$

За результатом вирішення цієї задачі симплекс-методом одержимо оптимальне розв'язання у вигляді

$$x_1 = 2,\ x_2 = 2,\ x_3 = 1,\ x_5 = 0;\ F_{opt} = 19.$$

*Приклад 2.*

Знайти $x_1, x_2, x_3, x_4$, що доставляють

$$\min\left\{ z = -\frac{7}{13} + \frac{x_1}{4} + \frac{4x_3}{3} \right\}$$

за умов: $x_1, x_2, x_3, x_4$ - цілі невід'ємні числа, $z$ - ціле число;

$$
\begin{aligned}
&2x_1 + x_2 + \frac{4x_3}{3} = \frac{13}{3};\\
&\frac{x_1}{4} + \frac{4x_3}{3} + x_4 = \frac{9}{4}.
\end{aligned}
$$

Якщо відкинути вимогу цілочисловості $x_j$, $j = \overline{1,4}$, оптимальне розв'язання задачі лінійного програмування можна вказати одразу: $\hat{x}_1 = 0$, $\hat{x}_2 = 13/3$, $\hat{x}_3 = 0$, $\hat{x}_4 = 9/4$, $\hat{z} = -7/12$, оскільки коефіцієнти функції мети при небазисних змінних додатні.

Значить, якщо покласти $\hat{x}_1 = y_1$, $\hat{x}_3 = y_3$, одержимо із початкових умов систему

$$
\begin{aligned}
&\hat{z} = -\frac{7}{12} + \frac{y_1}{4} + \frac{4y_3}{3};\\
&\hat{x}_1 = y_1;\ \hat{x}_3 = y_3;
\end{aligned}
$$

$$\hat{x}_2 = \frac{13}{3} - 2y_1 - \frac{4y_3}{3};$$

$$\hat{x}_4 = \frac{9}{4} - \frac{1}{2}y_1 - \frac{3}{4}y_3.$$

Вільні члени виразів $\hat{x}_2$ та $\hat{x}_4$ є дробові. Тому треба вводити у систему додаткове обмеження. Для формування такого обмеження зручно обрати вираз $\hat{z}$, оскільки у ньому всі коефіцієнти при $y_s > 0$ $(=1,3)$.

Позначимо нову змінну

$$\hat{x}_5 = -\frac{7}{12} + \frac{y_1}{4} + \frac{4y_3}{3}.$$

Отримуємо таку розширену систему

$$\hat{z} = -\frac{7}{12} + \frac{y_1}{4} + \frac{4y_3}{3}.$$

$$\hat{x}_1 = y_1;\ \hat{x}_3 = y_3;\ \hat{x}_2 = \frac{13}{3} - 2y_1 - \frac{4y_3}{3};$$

$$\hat{x}_4 = \frac{9}{4} - \frac{1}{2}y_1 - \frac{3}{4}y_3.\ \hat{x}_5 = -\frac{7}{12} + \frac{y_1}{4} + \frac{4y_3}{3}.$$

Оскільки у систему введено нову базисну змінну, треба вивести із базису одну зі змінних $\hat{x}_2 = 0$ або $\hat{x}_4 = 0$. Змінні $\hat{x}_1 = 0$ та $\hat{x}_3 = 0$ не можуть одночасно дорівнювати нулю (за такими їхніми значеннями $\hat{x}_5 < 0$).

Виведемо з базису змінну $y_2 = \hat{x}_2$, $y_5 = \hat{x}_5$. Отримаємо таку систему.

$$\hat{z} = y_5;\ \hat{x}_2 = y_2;\ \hat{x}_5 = y_5;$$

$$\hat{x}_1 = \frac{45}{21} - \frac{4}{7}y_2 - \frac{4}{7}y_5;\ \ \hat{x}_4 = \frac{129}{112} - \frac{23}{112}y_2 - \frac{5}{14}y_5;$$

$$\hat{x}_3 = \frac{1}{28} + \frac{3}{28}y_2 + \frac{6}{7}y_5.$$

Отже, базисне розв'язання

$$\hat{x}_1 = \frac{45}{21},\ \hat{x}_3 = \frac{1}{28},\ \hat{x}_4 = \frac{129}{112};\ \hat{x}_2 = \hat{x}_5 = 0,\ \hat{z} = 0$$

є новим оптимальним розв'язанням, яке отримано після відсічення першої точки $\hat{X}_1$. В отриманій системі, яка визначає точку $\hat{X}_{1,1}$ при $y_2 = y_5 = 0$ знову наявні дробові вільні члени, отже треба вводити нове додаткове обмеження. Для цього виберемо $\hat{x}_3$, в якому коефіцієнти при $y_2$ та $y_5$ додатні. $\hat{x}_6 = -\frac{27}{28} + \frac{3}{28}y_2 + \frac{6}{7}y_5$. Маємо нову розширену систему

$$\hat{z} = y_5;\ \hat{x}_2 = y_2;\ \hat{x}_5 = y_5;$$

$$\hat{x}_1 = \frac{45}{21} - \frac{4}{7}y_2 - \frac{4}{7}y_5;\ \hat{x}_4 = \frac{129}{112} - \frac{23}{112}y_2 - \frac{5}{14}y_5;$$

$$\hat{x}_3 = -\frac{1}{28} + \frac{3}{28}y_2 + \frac{6}{7}y_5;\ \hat{x}_6 = -\frac{27}{28} + \frac{3}{28}y_2 + \frac{6}{7}y_5.$$

Виводимо із базису $\hat{x}_1$. Отримуємо:

$$\hat{z} = y_5;\ \hat{x}_1 = y_1;\ \hat{x}_6 = y_6;$$

$$\hat{x}_2 = 3 - 2y_1 - \frac{4}{3}y_6;\ \hat{x}_4 = \frac{3}{2} - \frac{1}{2}y_1 - \frac{3}{4}y_6;$$

$$\hat{x}_3 = 1 + y_6;\ \hat{x}_5 = \frac{3}{4} + \frac{1}{4}y_1 + \frac{4}{3}y_6.$$

Знову отримане дробове оптимальне розв'язання. Додаткове обмеження: $\hat{x}_7 = -\frac{1}{4} + \frac{1}{4}y_1 + \frac{4}{3}y_6$. Вводимо в базис $\hat{x}_1$.

$$\hat{x}_1 = 1 - \frac{10}{3}y_6 - 4y_7;\ \hat{x}_5 = 1 + y_7;$$

$$\hat{x}_2 = 1 + \frac{28}{3}y_6 - 8y_7;\ \hat{x}_6 = y_6;$$

$$\hat{x}_4 = 1 + \frac{23}{12}y_6 - 3y_7;\ \hat{z} = 1 + y_7.$$

Отримане оптимальне розв'язання $x_1^* = 1,\ x_2^* = 1,\ x_3^* = 1,\ x_4^* = 1$ є цілочисловим.

## 2.5.3. Дискретне програмування

Розв'язанням задачі дискретного програмування називається вектор, координати якого належать області

$$\begin{aligned} &\sum_{j=1}^{n}\sum_{i=1}^{m} a_{ij}x_j && \le b_i,\ i = \overline{1,m}; \\ &x_j && \ge 0;\ j = \overline{1,n}; \\ &x_j \in A^j = (A_{j_1}, A_{j_2} && ,\ldots, A_{j_m}),\ j = \overline{1,n_1},\ n_1 \le n; \end{aligned} \tag{92}$$

і максимізують функцію

$$F(x) = \sum_{j=1}^{n} c_j x_j. \tag{93}$$

Передбачається, що $A_{j_i}$ проранжовано, тобто $0 = A_{j_1} < A_{j_2} < \cdots < A_{j_m}$.

**Теорема**. *Нехай* $x(G^k, F)$ *-- оптимальне розв'язання задачі лінійного програмування,* $x_{ij}^k$ *-- елементи симплексної таблиці, яка відповідає оптимальному розв'язанню.*

Якщо $x(G^k, F)$ є недопустимим розв'язанням задачі дискретного програмування і $A_{i\nu} < x_{i_0}^k < A_{i,]nu+1}$, тоді, використовуючи $i$-й рядок симплекс-таблиці, можна побудувати відсічення, що має властивість правильності, за формулами

$$z_i = -\gamma_0^k + \sum_{j \in N_k} \gamma_j^k x_j;\ z_i \ge 0, \tag{94}$$

де

$$\gamma_0^k = x_{i_0}^k - A_{i,\nu}$$

$$\gamma_j^k = \begin{cases} x_{ij}^k, & x_{ij}^k \geq 0, \\ \dfrac{x_{i_0}^k - A_{i,\nu}}{A_{i,\nu+1} - x_{i_0}^k}(-x_{ij}^k), & x_{ij}^k < 0. \end{cases} \qquad (95)$$

Згідно з цією теоремою алгоритм дискретного програмування (алгоритм Дальтона) може бути описаний так:

1. Вирішується $(G^k, F)$ -задача. Нехай $x(G^k, F)$ , $k = 0,1,2,\ldots$ -- оптимальне розв'язання $(G^k, F)$-задачі, $\| x_{ij}^k \|$ -- симплексна таблиця.

2. Перевіряється умова допустимості за всіма координатами оптимального розв'язання $x(G^k, F)$ $(G^k, F)$ -задачі. Якщо умова допустимості виконується, отримане розв'язання є оптимальним розв'язанням початкової задачі. Якщо ж умова допустимості розв'язання не виконується бодай за однією координатою, здійснюється перехід до п.3.

3. Нехай $x_{i+0}^k$ , $i \in (1, n_1)$ , не задовольняють умову допустимості. Тоді вибирається $i_0 = \min\{i \mid i \in (1, n_1)$ , $x_{i_0}^k$ не задовольняє (80).

4. Для вибраного номера $i = i_0$ будується правильне відсікання, тобто запроваджується додаткове обмеження (змінна)

$$x_{n+k+1} = -\gamma_0^k + \sum_{j \in N_k} (-\gamma_j^k)(-x_j), \qquad (96)$$

5. Додаємо лінійне обмеження, що визначає правильне відсікнення, до умов $(G^k, F)$ -задачі й отримуємо нову $(G^{k+1}, F)$-задачу. Привласнивши $k = k+1$, переходимо до п.1.

*Приклад*. Знайти $x = (x_1, x_2)$ , що максимізує $F(x) = x_1 + x_2$ за умов

$$\begin{aligned} -x_1 + x_2 &\leq 12, \\ 4x_1 - x_2 &\leq 12, \\ x \in \{0,1,4\}, &\quad x_2 \in \{0,1,3,5\}. \end{aligned}$$

Вирішуємо $(G^0, F)$-задачу

$$\begin{aligned} \max F(x) &= x_1 + x_2; \\ -x_1 + x_2 + x_3 &= 12, \\ 4x_1 - x_2 + x_4 &= 12, \\ x_j \geq 0, &\quad j = 1,2,3,4. \end{aligned}$$

Табл.2.10.

| Базис | | $x_1$ | $x_2$ | $x_3$ | $x_4$ |
|---|---|---|---|---|---|
| $x_3$ | 12 | 1 | 4 | 1 | 0 |
| $x_4$ | 12 | 4 | -1 | 0 | 1 |
| c | 0 | -1 | -1 | | |

Табл.2.11.

| Базис | | $x_1$ | $x_2$ | $x_3$ | $x_4$ |
|---|---|---|---|---|---|
| $x_2$ | 4 | 0 | 1 | 4/15 | 1/15 |
| $x_1$ | 4 | 1 | 0 | 1/15 | 4/15 |
| c | 8 | 0 | 0 | 1/3 | 1/3 |

Табл.2.12.

| Базис | | $x_1$ | $x_2$ | $x_3$ | $x_4$ | |
|---|---|---|---|---|---|---|
| Δ | 8 | 0 | 0 | 1/3 | 1/3 | 0 |
| $x_2$ | 4 | 0 | 1 | 4/15 | 1/15 | 0 |
| $x_1$ | 4 | 1 | 0 | 1/15 | 4/15 | 0 |
| $z$ | -1 | 0 | 0 | -4/15 | -1/15 | 1 |

Табл.2.13.

| Базис | | $x_1$ | $x_2$ | $x_3$ | $x_4$ |
|---|---|---|---|---|---|
| Δ | 27/4 | 0 | 9 | 9 | 1/4 |
| $x_2$ | 3 | 0 | 1 | 0 | 0 |
| $x_1$ | 15/4 | 1 | 0 | 1 | 1/4 |
| $x_3$ | 15/4 | 0 | 0 | 1 | 1/4 |

Отриманий оптимальний план не задовольняє умові допустимості за змінною $x_2$. Тому будуємо правильне відсічення

$$z = -1 - \frac{4}{15}x_3 - \frac{1}{15}x_4.$$

Вводимо в базис $x_3$ замість $z$.

Отримане розв'язання $x_1 = 15/4$, $x_2 = 3$ не задовольняє умові допустимості по $x_1$. Побудуємо правильне відсічення по рядку $x_1$. Маємо: $\gamma_0 = 11/4$, $\gamma_4 = -1/4$, тобто додаткове обмеження має вигляд $z = -11/4 - 1/4x_4$. В базис замість $z$ увійде $x_3$. Після виконання цієї ітерації отримаємо $x_1 = 1$, $x_2 = 3$, $F(P]x) = 4$.

## 2.6. Комбінаторні методи у дискретному програмуванні

Поміж задач цілочислового програмування значний інтерес являють задачі, в яких шуканий екстремальний розв'язок описується деякою перестановкою (комбінацією) відомого набору чисел. Ці задачі отримали назву *задач комбінаторного типу*.

До задач цього типу належить задача про призначення (персоналу), розв'язком якої є перестановка $(P_1, P_2, \ldots, P_n)$ чисел $1, 2, \ldots, n$. Кожне з призначень описується відповідністю: $P_i$ відповідає $i$ ($i = \overline{1, n}$). Але ця задача, що може бути записана у вигляді задачі лінійного програмування, є частинний випадок транспортної задачі і не викликає ускладнень з точки

зору методу її вирішення.

*Задача про покриття.* Маємо граф $G >$ Необхідно знайти мінімальний набір ребер графу, за якими вершина графу відповідає деякому ребру, що міститься у покритті.

Матриця $A = \| a_{ij} \|$ $(i = \overline{1,m}, \ j = \overline{1,n})$, має назву *матриці інціденцій*, якщо її елементи визначаються таким чином:

$$a_{ij} = \begin{cases} 1, & \text{вершина і відповідає ребру } j \\ 0, & \text{у решті випадків.} \end{cases}$$

Побудуємо математичну модель задачі. Нехай граф $G$ характеризується матрицею інціденцій $A$; $x_j$ -- булева змінна, що визначається як:

$$x_j = \begin{cases} 1, & \text{якщо ребро j увійде у покриття} \\ 0, & \text{у решті випадків.} \end{cases}$$

Необхідно мінімізувати

$$F = \sum_{j=1}^{n} x_j$$

за таких умов:

1) відповідності кожної вершини бодай одному ребру покриття

$$\sum_{j=1}^{n} a_{ij} x_j \geq 1, \ i = \overline{1,m}.$$

2) змінні $x_j$ -- булеві.

Прикладним аналогом цієї моделі може слугувати задача синтезу електронних схем:

-- побудова схеми, яка реалізує дану булеву функцію;

-- пошук найекономічнішої поміж усіх можливих схем.

*Задача про комівояжера.* Маємо $n$ міст. Від'їжджаючи з одного міста, комівояжер має об'їхати усі міста і повернутися у те саме місто. У кожне місто можна заїжджати тільки один раз, тому маршрут комівояжера утворює замкнутий цикл без петель. Відома матриця транспортних відстаней, при цьому кожне місто з'єднане з іншими. Необхідно відшукати мінімальний замкнутий маршрут.

Ця задача відрізняється від задачі про призначення у тому відношенні, що необхідно побудувати замкнутий маршрут. Побудуємо математичну модель задачі.

$$x_{ij} = \begin{cases} 1, & \text{якщо комівояжер із міста і переїжджає у місто } j, \\ 0, & \text{у решті випадків,} \end{cases}$$

де $i, j = \overline{1,n}$, $i \neq j$.

Потрібно мінімізувати

$$F(x) = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij} \qquad (97)$$

за умови

$$\sum_{i=1}^{n} x_{ij} = 1, \ j = \overline{1,n}, \qquad (98)$$

$$\sum_{j=1}^{n} x_{ij} = 1,\ j = \overline{1,n}, \tag{99}$$

$$u_i - u_j + n x_{ij} \le n-1,\ i, j = \overline{1,n}, i \ne j \tag{100}$$

де $u_i, u_j$ довільні значення.

Умова (85) забезпечує замкнутість маршруту та відсутність петель.

### 2.6.1. Метод гілок та меж

Метод гілок та межрозглянемо на прикладі задачі про комівояжера. Нехай $C = \| c_{ij} \|$ -- матриця, елементи якої визначають витрати при переході із міста $i$ у місто $j$; ці міста утворюють пару $(i, j)$.

*Циклом* $t$ назвемо набір із $n$ упорядженних пар міст, що утворюють маршрут, який проходить через кожне місто тільки один раз

$$t = [(i_1, i_2), (i_2, i_3(, \ldots, (i_{n-1}, i_n), (i_n, i_1)] \tag{101}$$

Кожна пара $(i, j)$ утворює комунікацію маршруту. Тоді загальні витрати $z(t)$ для циклу $t$, що відповідають матриці $C$, це сума елементів матриці по комунікаціям маршруту

$$z(t) = \sum_{(i,j) \in t} c_{ij}.$$

Запровадимо поняття приведеної матриці та процесу приведення. Якщо з усіх елементів деякого $i$-го рядку або $j$-го стовпця матриці $C$ відняти найменший з них, отримаємо деяку матрицю, що має властивість -- у кожному її рядку та у кожному її стовпцю щонайменше один нуль. Така матриця зветься *приведеною*, а процес утворення нулів -- приведенням. Сума елементів, що віднімається у процесі приведення,зветься *сталими, що приводять* і позначається $h^k$, де $k$ -- порядковий номер приведення. Вочевидь, матриця $C$ містить невід'ємні елементи. Перехід до приведеної атриці організовано таким чином, що усі її елементи також невід'ємні. Побудуємо процес пошуку оптимального маршруту для приведених матриць таким чином, щоб він збігався із оптимальним маршрутом початкової задачі із матрицею $C$.

Нехай $c_{i,j(i)} = \min_i c_{ij}$, тоді

$$c'_{i,j} = c_{ij} - c_{i,j(i)}. \tag{102}$$

Нехай $c^*_{i(j),j} = \min_i c'_{ij}$, тоді

$$c''_{i,j} = c'_{ij} - c^*_{i(j),j}. \tag{103}$$

Такий процес надає можливість з невід'ємної матриці $C$ отримати приведену матрицю $C''$ із сумою сталих, що приводять,

$$h^1 = \sum_{i=1}^{n} c_{i,j(i)} + \sum_{j=1}^{n} c^*_{i(j),j} \tag{104}$$

Якщо $z(t)$ -- витрати циклу $t$ для матриці до приведення, $z^1(t)$ -- витрати циклу $t$ для матриці після приведення, $h^1$ -- сума сталих, що приводять,

$$z(t) = h^1 + z^1(t). \tag{105}$$

Оскільки приведена матриця містить тільки невід'ємні елементи, $h^2$ є нижня межа

витрат циклу $t$ зі старою матрицею.

Для опису процедури розгалуження використаємо геометричну інтерпретацію цього методу. Поділ множини усіх циклів на підмножини, що не перетинаються, подамо у вигляді дерева гілок з вершинами, що складаються з пар допустимих $(i, j)$ та заборонених $\overline{(i, j)}$ переїздів. Отже, гілка, що проходить через вершину $(i, j)$, містить усі маршрути, які містять переїзд із $i$ у $j$. Гілка, що проходить через $\overline{(k,l)}$, містить усі маршрути, в яких здійснюється переїзд з міста $i$ у місто $j$ і заборонено переїзд із $k$ у $l$. У той самий час гілка, що проходить через $(k,l)$, містить усі маршрути, в яких слідом за містом $i$ відвідується місто $j$, а слідом за містом $k$ -- місто $l$.

Якщо із деякої вершини $X$ пройти по дереву вгору, можна назвати усі вершини, які входять у цикли, що належать $X$, а також ті, що не можуть входити у ці цикли.

Позначимо вершину, з якої здійснюється розгалуження, через $X$, вершину, яка, найімовірніше, належить гілці, що є перспективною, позначимо $Y$, а вершину, що заборонена, -- $\bar{Y}$. співставимо у відповідність до вершини $X$ суму сталих, що приводять, $h^k$, які являють собою нижню межу $\omega(X)$.

Організуємо процесу вибору пари міст для розгалуження, тобто побудову множин $\bar{Y}$ та $Y$, як гру двох осіб. Гравець $Y$ має таку перевагу: він може обирати будь-яку з пари міст, що не обирались до об'їзду. Вочевидь, щоб побудувати цикл мінімальної довжини, він має обирати ті з пар міст, яким у приведеній матриці відповідають нульові елементи. Оскільки у кожному рядку та стовпці приведеної матриці міститься бодай один нульовий елемент, кандидатів для включення в $Y$ на першому кроці буде не менше за $n$. У зв'язку з цим для вибору однозначної стратегії перший гравець накладає обмеження на поведінку другого гравця $\bar{Y}$:

-- якщо пара $i, j)$ обирається гравцем $y$, для $\bar{Y}$ з міста $i$ треба виїхати, але не в $j$, а у будь-яке інше місто; в $j$ треба виїхати, але не з $i$, а з будь-якого іншого міста. За такої ситуації не виключено, що збитки $\bar{Y}$ виявляться більшими за збитки, пов'язані із безпосереднім переїздом з $i$ у $j$. Вочевидь, $\bar{Y}$ прагнутиме мінімізувати збитки, тож у $i$-му рядку він обере місто, якому відповідає найменша ``відстань'' (крім $j$), Аналогічно щодо $j$.

Гравець $Y$ обирає серед усіх допустимих пар ту пару, яка призводить до найбільших витрат $\bar{Y}$. Запишемо алгоритм:

1. $k = 1$.

2. Здійснимо приведення матриці $C$ і створимо приведену матрицю $C^k$.

3. Обчислимо суму сталих $h^k$ -- оцінку для множини $X$ -- позначимо її $\omega(X)$.

4. Оберемо претендентів на включення у множину $Y$, тобто усі пари $(, j)$, $i \neq j$, для яких $c_{ij}^k = 0$.

5. Обчислимо для цих претендентів

$$\theta(k,l) = \min_{j' \neq j} c_{i,j'} + \min_{i' \neq i} c_{i',j}.$$

6. Оберемо

$$\theta(i, j) = \max_{i,j);c_{i,j}^k = 0} \theta(i, j).$$ Пара $(k,l) \to Y$ й є заборонена для $\bar{Y}$.

7. Підрахуємо оцінку для множини $\bar{Y}$:

$$\omega(\bar{Y}) = \omega(X) + \theta(k,l).$$

8. Оскільки із кожного міста можна виїжджати тільки один раз, природно $k$-й рядок та $l$-й стовпець виключити з подальшого аналізу. Для запобігання утворення замкнутих підциклів треба заборонити переїзд із $l$ у $k$, тобто клітинку $(k,l)$.

9. Матриця, що отримується на деякому кроці, має розмірність $2\times 2$ і містить лише дві допустимі пари міст. Ці пари є замикаючі для деякого маршруту без петель. Отже, момент утворення матриці $2\times 2$ є ключовий. Якщо матриця має розмір $2\times 2$, переходимо до п. 11, якщо ні -- до п. 10.

10. Чи є отримана матриця приведеною? Якщо так, оцінка для множини $Y$ дорівнює оцінці множини, з якої отримано $Y$, тобто $\omega(Y) = \omega(X)$. якщо ні -- здійсн7юється приведення матриці, яку щойно отримано, та обчислюються $h^{(k+1)}$, після чого

$$\omega(Y) = \omega(X) + h^{(k+1)},\ k = k+1.$$

Переходимо до п. 4.

11. Переверка умови оптимальності: якщо оцінка замкнутого циклу не більше за оцінки усіч допустимих для подальшого розгалуження (кінцевих на гілках) множин, отриманий замкнутий маршрут є оптимальний. Якщо ж існує бодай одна множина, для якої оцінка є менша, отриманий замкнутий маршрут запам'ятовується, а процес розгалуження триває, виходячи із множин із меншою оцінкою.

## 2.7. Апроксимуюче лінійне програмування

Сформулюємо задачу нелінійного програмування

мінімізувати $f(x)$, $x \in E^n$,

з обмеженнями

$$\begin{aligned} &h_i(x) = 0, \quad i = \overline{1,m}, \\ &g_j(x) \ge 0, \quad j = \overline{m+1,p}. \end{aligned} \tag{106}$$

Методи лінійного програмування з успіхом застосовуються до розв'язання задач великої розмірності із обмеженнями як у вигляді рівностей, так і у вигляді нерівностей лінійної структури. Цілком слушно спробувати ці методи застосувати до розв'язання задач нелінійного програмування шляхом лінеаризації за належним чином побудованого ітераційного процесу.

Лінійна апроксимація нелінійних функцій, наявних у задачі (91), досягається заміною функцій членами першого порядку у відповідному розкладанні у ряд Тейлора в околі точки $x^{(k)}$. Внаслідок процесу лінеаризації нелінійних функцій, який повторюється в околі кожного проміжного розв'язання шляхом розкладанні у ряд Тейлора, створюється послідовність $x^{(0)}, x^{(1)}, \ldots, x^{(k)}$, яка за визначеними умовами збігається до оптимального розв'язку $x^*$ початкової задачі нелінійного програмування.

Нехай нам відома точка $x^{(k)}$. Розкладемо в околі цієї точки нелінійн функції $f(x^{(k)})$, $h_i(x^{(k)})$, $g_j(x^{(k)})$ у ряд Тейлора. Отримаємо таку задачу:

мінімізувати

$$f(x^{(k)}) + \nabla^T f(x^{(k)})(x - x^{(k)}),\ x \in E^n, \tag{107}$$

із обмеженнями

$$\begin{aligned} h_i(x^{(k)}) + \nabla^T h_i(x^{(k)})(x - x^{(k)}) \quad &= 0,\ i = \overline{1, m}, \\ g_j(x^{(k)}) + \nabla^T g_j(x^{(k)})(x - x^{(k)}) \quad &\geq 0,\ j = \overline{m+1, p}. \end{aligned} \tag{108}$$

Оскільки $f(x^{(k)})$, $h_i(x^{(k)})$, $g_j(x^{(k)})$, $\nabla^T f(x^{(k)})$, $\nabla^T h_i(x^{(k)})$, $\nabla^T g_j(x^{(k)})$ є або скалярні сталі, або сталі вектори, значення яких обчислюються у точці $x^{(k)}$, задача (92), (93) являє собою задачу лінійного програмування.

З метою ілюстрації методу лінеаризації задачі нелінійного програмування в околі деякої точки $x^{(k)}$ розглянемо таку задачу:

мінімізувати

$$f(x) = 4x_1 - x_2^2 - 12$$

з обмеженнями

$$\begin{aligned} h_1(x) = \quad & 25 - x_1^2 - x_2^2 = 0, \\ g_2(x) = \quad & 10x_1 - x_1^2 + 10x_2 - x_2^2 - 34 \geq 0, \\ g_3(x) = \quad & x_1 \geq 0, \\ g_4(x) = \quad & x_2 \geq 0. \end{aligned}$$

Нехай $x^{(k)} = [2\ 4]^T$. Замістимо нелінійні функції на лінійні вирази, що їх апроксимують. Отримаємо таку задачу лінійного програмування:

мінімізувати

$$\tilde{f}(x^{(k)}) = 4x_1 - 8x_2 + 4$$

із обмеженнями

$$\begin{aligned} \tilde{h}_1(x) = \quad & 45 - 4x_1 - 8x_2 = 0, \\ \tilde{g}_2(x) = \quad & -14 + 6x_1 + 2x_2 \geq 0, \\ \tilde{g}_3(x) = \quad & x_1 \geq 0, \\ \tilde{g}_4(x) = \quad & x_2 \geq 0. \end{aligned}$$

Збіжність до вихідного розв'язання задачі (92), (93) гарантується, якщо виконуються такі умови:

1. Нелінійні функції неперервні і диференційовні.

2. Функція $f(x)$ випукла, сума $\sum h_i^2(x)$ теж випукла в $R$.

3. Функції $g_{m=1}(x), \ldots, g_p(x)$ -- увігнуті функції.

4. Допустима область не порожня, тобто існують такі значення складових вектора $c$, для яких задовольняються усі умови, що визначаються обмеженнями задачі (існує розв'язок).

5. Множина $R$ замкнута і випукла.

6. Існує $\varepsilon > 0$ таке, що $|h_i(x)| \leq \varepsilon$ та $|g_i(x)| \geq -\varepsilon$, тобто функції, наявні в умовах задачі, обмежені.

### 2.7.1. Метод апроксимуючого програмування

Метод апроксимуючого лінійного програмування (МАП) грунтується на багаторазовому застосуванні алгоритмів лінійного програмування, що забезпечує на межі

збіжність послідовності розв'язків проміжних задач лінійного програмування до розв'язків задачі нелінійного програмування.

Після застосування процедури лінеаризації задача формулюється так:

мінімізувати

$$f(x) - f(x^{(k)}) = \sum_{j=1}^{n} \frac{\partial f(x^{(k)})}{\partial x_j} \Delta x_j^{(k)} \tag{109}$$

за умов, що мають місце обмеження у вигляді рівностей

$$\sum_{j=1}^{n} \frac{\partial h_i(x^{(k)})}{\partial x_j} \Delta x_j^{(k)} = -h_i(x^{(k)}), \; i = \overline{1, m}, \tag{110}$$

та обмеження у вигляді нерівностей

$$\sum_{j=1}^{n} \frac{\partial g_j(x^{(k)})}{\partial x_j} \Delta x_j^{(k)} = -g_j(x^{(k)}), \; i = \overline{m+1, p}, \tag{111}$$

де $\Delta x_i^{(k)} = x_i - x_i^{(k)}$.

Метод апроксимуючого програмування надає можливість враховувати усі обмеження, які записано у вигляді нерівностей.

МАП реалізується за такою схемою. Нехай $x^{(k)}$ -- допустима точка в $R$. Після апроксимації нелінійних функцій в околі точки $x^{(k)}$ шукається розв'язок задачі лінійного програмування із додатковою умовою

$$\delta_j^{k)} - | \, x_j^{(k+1)} - x_j^{(k)} \, | \geq 0, \;\; j = \overline{1, n}, \tag{112}$$

де $\delta_j^{(k)} > 0$ -- мала величина, що обмежує величину кроку і не дозволяє вектору $x$ виходити за межі допустимої області задачі (94)-- (96).

Позначимо $\Delta^+ x^{(k)} = x - x^{(k)}$ за $x \geq x^{(k)}$ та $\Delta^- x^{(k)} = x - x^{(k)}$ за $x \leq x^{(k)}$. Тоді перед тим як реалізувати чергову ітерацію, слід прийняти до уваги обмеження для допустимих переміщень у просторі розв'язань, які визначаються співвідношеннями:

$$p_j^{(k)} \Delta^+ x_j^{(k)} + q_j^{(k)} \Delta^- x_j^{(k)} \leq m_j^{(k)}, \;\; j = \overline{1, n}, \tag{113}$$

де

$$p_j^{(k)} = \max\left\{1, \frac{m_j^{(k)}}{U_j - x_j^{(k)}}\right\}, \;\; q_j^{(k)} = \max\left\{1, \frac{m_j^{(k)}}{x_j^{(k)} - L_j}\right\},$$

$m_j^{(k)}$ -- максимально допустиме переміщення уздовж $j$-ї осі координат на $k$-му кроці; $L_j$ -- нижня межа для $x_j$; $U_j$ -- верхня межа для $x_j$.

Якщо $\Delta^- x_j^{(k)} = 0$, $p_j^{(k)} \Delta^+ x_j^{(k)} \leq m_j^{(k)}$. Коли значення $x_j^{(k)}$ близьке до своєї верхньої межі, тобто $U_j - x_j^{(k)} \approx 0$, максимальним виявляється $m_j^{(k)} / (U_j - x_j^{(k)})$. Отже, $x_j - x_j^{(k)} \leq U_j - x_j^{(k)}$, або $x_i \leq U_j$. Тобто є гарантія, що $x_j$ не може перевищити своє верхнє межове значення. Аналогічні міркування стосовно умов, коли $\Delta^+ x_j^{(k)} = 0$, дозволяють пересвідчитися, що $x_j$ не може приймати значення, менші за $L_j$.

Співвідношення (94)-- (98) утворюють систему лінійних співвідношень, яка й підлягає аналізу за допомогою методів лінійного програмування. Зазвичай, перший крок у процесі пошуку розв'язання даної задачі лінійного програмування за допомогою симплекс-методу

полягає у запровадженні у розгляд додаткових, *послаблюючих та штучних змінних* . Використовуючи для позначення послаблюючих змінних символ $u_i$ , можна записати обмеження у вигляді нерівностей у такій формі:

$$\sum_{j=1}^{n}\frac{\partial g_j(x^{(k)})}{\partial x_j}\Delta^{+}x_j^{(k)}-\sum_{j=1}^{n}\frac{\partial g_j(x^{(k)})}{\partial x_j}\Delta^{-}x_j^{(k)}-u_i^{(k)}=-g_j(x^{(k)}),\ i=\overline{m+1,\ p},. \qquad (114)$$

Для перетворення у рівності обмежень (98) також використовуються послаблюючі змінні $v_i$ :

$$p_j^{(k)}\Delta^{+}x_j^{(k)}+q_j^{(k)}\Delta^{-}x_j^{(k)}+v_j^{(k)}=m_j^{(k)},i=\overline{1,n}. \qquad (115)$$

Щоб мати можливість завдання як початкової допустиму точку у разі, коли початковий вектор $x$ виявиться за межами допустимої області, запроваджуються штучні змінні $w_i$ . Отже, отримуємо систему обмежень у вигляді

$$\sum_{j=1}^{n}\frac{\partial h_i(x^{(k)})}{\partial x_j}\Delta^{+}x_j^{(k)}-\sum_{j=1}^{n}\frac{\partial h_i(x^{(k)})}{\partial x_j}\Delta^{-}x_j^{(k)}+w_i^{(k)}=-h_i(x^{(k)}),\ i=\overline{1,m}, \qquad (116)$$

$$\sum_{j=1}^{n}\frac{\partial g_i(x^{(k)})}{\partial x_j}\Delta^{+}x_j^{(k)}-\sum_{j=1}^{n}\frac{\partial g_i(x^{(k)})}{\partial x_j}\Delta^{-}x_j^{(k)}-u_i^{(k)}+w_i^{(k)}=-g_i(x^{(k)}),\ i=\overline{m+1,\ p}. \qquad (117)$$

Штучні змінні не дорівнюють нулю, якщо умова, що визначається відповідним обмеженням, не задовольняється. Застосування модифікованого симплекс-методу надає можливість перетворити кожну зі штучних змінних на нуль шляхом мінімізації $\sum w_i$ .

У ході лінійного програмування спочатку прагнуть задовольнити обмеження, що не дозволяють вектору $x$ виходити за межі допустимої області, а потім прагнуть покращити значення функції мети. При цьому не виключається можливість появи такого вектору $x$ , що хоча він і забезпечує мінімізуючу поправку $f(x)$ , але виявиться за межами допустимої області. При цьому у зв'язку з тим, що на кожному кроці виконується повна релінеаризація задачі, вся попередня інформація втрачає сенс. З цієї причини МАП може застосовуватися для розв'язання невипуклих задач.

*Приклад.* Нехай потрібно мінімізувати

$$f(x)=x_1^2+x_2^2-16x_1-10x_2=(x_1-8)^2+(x_2-5)^2-89$$

з обмеженнями

$$g_1(x)=\ 11-x_1^2+6x_1-4x_2\ge 0,$$

$$g_2(x)=\ x_1x_2-3x_2-e^{x_1-3}+1\ge 0,$$

$$g_3(x)=\ x_1\ge 0,$$

$$g_4(x)=\ x_2\ge 0.$$

Графіки функції мети та обмежень наведено на рис.2.4.

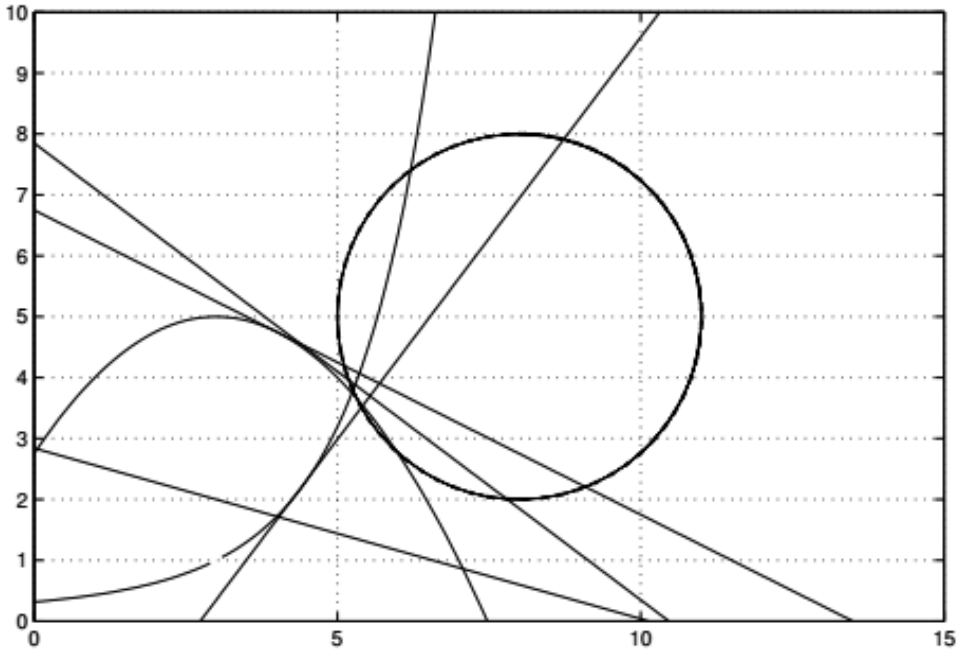


Рис. 2.4. Метод апроксимуючого програмування

Осі координат відповідають останнім двом обмеженням. Пунктирними лініями позначено лінії рівня функції мети -- кола із центром у точці $(8,5)$.

Як початкову виберемо точку, яка належить області допустимих значень (задовольняє усім обмеженням). Нехай це буде точка $x(0) = [4\ 3]^T$. У цій точці $F(|\ bfx^{(0)}) = -69$.

Лінійна апроксимація задачі в околі цієї точки призводить до задачі:
мінімізувати

$$\tilde{f}(x^{(0)}) = -8x_1 - 4x_2 - 25$$

з обмеженнями

$$\begin{aligned} g_1(x^{(0)}) = &\quad -2x_1 - 4x_2 + 27 \geq 0, \\ g_2(x^{(0)}) = &\quad 0,28x_1 + x_2 - 2,84 \geq 0, \\ g_3(x^{(0)}) = &\quad x_1 \geq 0, \\ g_4(x^{(0)}) = &\quad x_2 \geq 0. \end{aligned}$$

Розв'язанням цієї задачі є вектор $x^{(1)} = [13,5\ 0]^T$. Він є допустимий відносно лінеаризованої задачі, але не є допустимий відносно початкової задачі.

Щоб запобігти виходу вектора $x$ із допустимої області початкової задачі, на $x$ накладають обмеження вигляду ((112)) за припущення, що

$$|\ x_j^{(1)} - x_j^{(0)}\ | \leq \delta_j^{(0)},\ \ i = \overline{1,n}. \qquad (118)$$

Для нашої задачі початкове значення $\delta(0) = [0,5 - 0,5]^T$. Тоі отримуємо

$$x^{(1)} = [(4 + 0,5)\ (3 - 0,5)]^T = [4,5\ 2,5]^T;\ f(x^{(1)}) = -70,65.$$

На наступному кроці ($k = 1$) припустимо $\delta^{(1)} = 0,86\delta^{(0)} = [0,4 - 0,4]^T$. Лінеаризована задача для $x^{(1)}$ набуває вигляду

мінімізувати

$$\tilde{f}(x^{(1)}) = -7x_1 - 5x_2 - 36$$

із обмеженнями

$$\begin{aligned}
\tilde{g}_1(x^{(0)}) &= \quad 31,4 - 3x_1 - 4x_2 \geq 0,\\
\tilde{g}_2(x^{(0)}) &= \quad 5,42 - 1,98x_1 + 1,5x_2 \geq 0,\\
\tilde{g}_3(x^{(0)}) &= \quad x_1 \geq 0,\\
\tilde{g}_4(x^{(0)}) &= \quad x_2 \geq 0.
\end{aligned}$$

Розв'язання цієї задачі є точка $x^{(2)} = [5,45\,3,58]^T$. Якщо накласти додактове обмеження $x_j^{(2)} - x_j^{(1)} | \leq \delta_j^{(1)}$, як допустиме розв'язання отримаємо $x^{(2)} = [4,9\,2,9]^T$, $f(x^{(2)}) = -75,1$, $\delta^{(2)} = 0,8\delta^{(1)} = [0,32 - 0,32]^T$. Ітераційний процес триває, доки не справдиться умова $|\delta^{(k+1)}| \leq \varepsilon$.

**Труднощі**, що виникають при розв'язанні задач нелінійного програмування шляхом багаторазової лінеаризації та використання апарату лінійного програмування:

1. Оптимізація виконується повільно, якщо прагнути зробити вектори $x^{(k-1)}$, $x^{(k)}$,… допустимими або дуже близькими до допустимих.

2. Важко порівнювати два вектори $x$, що слідують один за одним і які призводять до різних значень $f(x)$ та із різним ступенем порушують деяку підмножину обмежень.

3. Якщо функція мети характеризується високим ступенем нелінійності, на грунті розв'язання лінеаризованої задачі напрям пошуку оптимальної точки може бути вибраний надто неточно. Лінійне програмування не дозволяє правильно обирати напрям пошуку також у тому випадку, коли поверхня, що обмежує допустиму область, занадто розтягнута або має ребристу структуру (порушується умова неперервності функцій).

4. Крім того, варто відмітити, що при застосуванні методу апроксимуючого програмування важко придумати таку процедуру урахування апріорних даних, яка б належним чином прискорювала процес оптимізації.

## 2.8. Квадратичне програмування

З метою запобігання труднощів, викладених вище, функцію мети апроксимують квадратичною формою. У разі розв'язання оптимізаційних задач без обмежень цей метод переходить у метод найскорішого спуску.

Функція мети задачі квадратичного програмування дорівнює

$$F(x) = c^T x + x^T Dx, \tag{119}$$

де $x$ -- $n$-вимірний вектор змінних, $c$ -- $n$-вимірний вектор сталих, $D$ -- $n \times n$-матриця сталих.

Лінійні обмеження запишемо у матричному вигляді

$$Ax = b,\ x_i \geq 0,\ i = \overline{1,n}, \tag{120}$$

$b$ -- $m$-вимірний вектор сталих, $A$ -- $m \times m$-матриця сталих.

У такій формі запис вектора $x$ може мати додаткові змінні, якщо деякі з початкових обмежень є нерівності. Рівняння (105) можна записати у формі

$$g(x) = Ax - b = 0. \tag{121}$$

Тоді задачу квадратичного програмування можна записати у такій компактній формі:

$$\min\{c^T x + x^T Dx \mid g(x) = 0;\ x_i \geq 0, i = \overline{1,n}\}. \tag{122}$$

Отже, для того, щоб квадратична форма (107) була випуклою, матриця $D$ має бути додатно піввизначеною. Це означає, що якщо

$$x^T Dx \geq 0 \ \forall x,$$

для будь-яких $x_1$, $x_2$ має виконуватися нерівність

$$[\lambda x_2 + (1-\lambda)x_1]^T D[\lambda x_2 + (1-\lambda)x_1] \leq \lambda x_2^T Dx_2 + (1-\lambda)x_1^T Dx_1 \tag{123}$$

за будь-якого $0 \leq \lambda \leq 1$.

За цих умов випуклість гарантує наявність глобального екстремуму.

Розглянемо деякі алгоритми, що вирішують задачі квадратичного програмування.

## 2.8.1. Метод Вульфа

Згідно теоремі Куна--Такера запишемо функцію Лагранжа $L(x,u)$ задачі квадратичного програмування ((122)):

$$L(x,u) = c^T x + x^T Dx + u^T g(x) \tag{124}$$

або у скалярній формі

$$L(x,u) = \sum_{i=1}^{n} c_i x_i + \sum_{i=1}^{n}\sum_{j=1}^{m} x_i d_{ij} x_j + \sum_{j=1}^{m} u_j g_j(x). \tag{125}$$

Оскільки усі обмеження $g_j(x)$ є рівності, доданої задачі мінімізації можна застосувати метод, аналогічний методу множників Лагранжа, роль яких тут відіграють $u_i$. Маємо

$$\frac{\partial L(x,u)}{\partial x} = c + 2Dx + A^T u \geq 0. \tag{126}$$

Застосуємо

$$\frac{\partial L(x,u)}{\partial u_j}\Big|_{x^0,u^0} = 0, \quad j = \overline{1,m} \tag{127}$$

до (109). Знову отримаємо рівності-обмеження у вигляді (105). Запровадимо тепер нову додаткову змінну

$$v = \frac{\partial L}{\partial x}. \tag{128}$$

Тепер можна переписати задачу квадратичного програмування у вигляді такої еквівалентної задачі.

Знайти вектор $x$, що задовольняє умови

$$Ax = b, \tag{129}$$

$$2Dx - v + A^T u = -c, \tag{130}$$

$$x^T v = 0,\ x_i \geq 0,\ v_i \geq 0, i = \overline{1,n}. \tag{131}$$

Із теореми Куна--Такера випливає, що вектор $x^0$ як розв'язок системи рівнянь (114)--(116) є також оптимальним розв'язком задачі квадратичного програмування (107). Оскільки

$x_i v_i = 0$, тільки $n$ із $2n$ змінних $x, v$ можуть відрізнятися від нуля. Це означає, що $n+m$ рівнянь (114), (115) можуть мати тільки $n+m$ змінних, що не дорівнюють нулю. Отже, цей єдиний розв'язок є базисний. Для визначення базисного розв'язку рівнянь (114)-- (116) можна застосувати симплексний метод лінійного програмування.

## 2.9. Двоїстість у задачах ЛП

ІЗ кожною задачею ЛП пов'язана інша цілком визначена задача ЛП. Зв'язок між цими задачами взаємний (тобто якщо первісною вважати другу задачу, то їй відповідає перша і навпаки). Цей зв'язок настільки тісний, що при розв'язанні однієї задачі фактично розв'язується інша. Ця пара задач має назву взаємно двоїстих (взаємно сполучених) задач.

Двоїстість у задачах ЛП полягає у тому, що кожній задачі відповідає двоїста до неї задача, розв'язання якої дає можливість отримати величини, які надають можливість виконати корисний аналіз первісної задачі ЛП.

Нехай дано канонічну задачу ЛП

$$c_1x_1 + c_2x_2 + \cdots + c_nx_n \to \max; \tag{132}$$

$$Ax = b,\ x_i \geq 0,\ i = \overline{1,n}. \tag{133}$$

*Двоїстою* до неї називається задача

$$b_1z_1 + b_2z_2 + \cdots + b_mz_m \to \min; \tag{134}$$

$$Az \geq c. \tag{135}$$

Задача, що є двоїста до канонічної задачі ЛП, будується за простим правилом: вільні члени вихідної задачі перетворюються на коефіцієнти функції мети, вектори-стовпці коефіцієнтів біля невідомих задачі (117) -- у вектори-рядки коефіцієнтів біля невідомих задачі ((134)); коефіцієнти функції мети задачі (119) -- у вільні члени задачі (117) .

*Приклад.* Побудувати задачу, двоїсту до задачі

$$\begin{aligned}
&3x_1 + 2x_2 - x_3 && \to \max;\\
&2x_1 - 2x_2 + x_3 - 4x_4 && = 8;\\
&x_1 + 4x_2 - 2x_3 + x_4 && = 4;\\
&2x_1 - x_2 - x_3 && = 2;\\
&x_j \geq 0, && j = \overline{1,4}.
\end{aligned}$$

Отримаємо двоїсту задачу у вигляді

$$\begin{aligned}
&8z_1 + 4z_2 + 2z_3 && \to \min;\\
&2_1 + z_2 + 2z_3 && \geq 3;\\
&-2z_+ 4z_2 - z_3 && \geq 2;\\
&z_1 - 2z_2 - z_3 && \geq -1;\\
&-3z_1 + z_2 && \geq 0..
\end{aligned}$$

У матричній формі задачі (117) і (119) записуються у вигляді

$$cx \to \max;$$

$$Ax = b,\ x \geq 0;$$

$$b^T z \to \min;$$

$$A^T z \geq c^T.$$

## 2.10. Завдання для самостійної роботи

1. Знайти розв'язки наступних задач ЛП графічним методом та симплекс-методом.
2. Записати двоїсті до них задачі.

2.1.

$$\begin{aligned} f = 7x_1 + 6x_2 &\to \max; \\ 2x_1 + 5x_2 &\geq 10; \\ 5x_1 + 2x_2 &\geq 10; \\ x_1 &\leq 6; \\ x_2 &\leq 5, \\ x_1, x_2 &\geq 0. \end{aligned}$$

2.2.

$$\begin{aligned} f = 5x_1 - 3x_2 &\to \min; \\ 3x_1 + 2x_2 &\geq 6; \\ 2x_1 - 3x_2 &\geq 10; \\ x_1 - x_2 &\leq 4; \\ 4x_1 + 7x_2 &\leq 26; \\ x_1, x_2 &\geq 0. \end{aligned}$$

2.3.

$$\begin{aligned} f = 3x_1 - 2x_2 &\to \max; \\ 2x_1 + x_2 &\leq 11; \\ -3x_1 + 2x_2 &\leq 10; \\ 3x_1 + 4x_2 &\geq 20; \\ x_1, x_2 &\geq 0. \end{aligned}$$

2.4.

$$\begin{aligned} f = x_1 + 2x_2 &\to \min; \\ -3x_1 + 2x_2 &\leq 9; \\ 3x_1 + 4x_2 &\geq 27; \\ 2x_1 + x_2 &\leq 14; \\ x_1, x_2 &\geq 0. \end{aligned}$$

2.5.

$$\begin{aligned} f &= 7x_1 - 2x_2 && \to \max; \\ 5x_1 - 2x_2 &&& \le 3; \\ x_1 + x_2 &&& \ge 1; \\ -3x_1 + x_2 &&& \le 3; \\ 2x_1 + x_2 &&& \le 4; \\ x_1, x_2 &&& \ge 0. \end{aligned}$$

2.6.

$$\begin{aligned} f &= -2x_1 + x_2 && \to \min; \\ 2x_1 + x_2 &&& \le 8; \\ x_1 + 3x_2 &&& \ge 6; \\ 3x_1 + x_2 &&& \ge 3; \\ x_1, x_2 &&& \ge 0. \end{aligned}$$

3. Знайти розв'язки таких задач ЛП.

3.1.

$$\begin{aligned} f &= -2x_3 + 2x_4 + x_5 && \to \max; \\ x_1 - 2x_2 + 2x_3 + x_5 &&& = 4; \\ x_2 + 3x_3 - x_4 + 3x_5 &&& = 2; \\ x_j &&& \ge 0,\ j = \overline{1,5}. \end{aligned}$$

3.2.

$$\begin{aligned} f &= x_1 + x_2 + 3x_3 + 8x_4 && \to \min; \\ 2x_1 - x_2 + 3x_3 - 2x_4 &&& = 3; \\ -x_1 + 3x_2 - 4x_3 + 4x_4 &&& = 1; \\ x_j &&& \ge 0,\ j = \overline{1,4}. \end{aligned}$$

3.3.

$$\begin{aligned} f &= x_1 + x_2 + 3x_3 + 8x_4 && \to \min; \\ 2x_1 - x_2 + 2x_3 + 8x_4 &&& = 3; \\ -x_1 + 3x_2 - 4x_3 + 4x_4 &&& = 1; \\ x_j &&& \ge 0,\ j = \overline{1,4}. \end{aligned}$$

3.4.

$$\begin{aligned} f &= 2x_1 - x_2 + 3x_3 + 2x_4 && \to \max; \\ x_1 - 2x_2 + 5x_3 - 3x_4 - x_6 &&& = 1; \\ 2x_1 + 3x_2 x_3 + 2x_4 + x_5 &&& = 4; \\ 4x_1 + 10x_2 + 3x_3 + x_4 + x_7 &&& = 8; \\ x_j &&& \ge 0,\ j = \overline{1,7}. \end{aligned}$$

3.5.

$$
\begin{aligned}
f &= 3x_1 - 2x_2 && \to \max;\\
2x_1 + x_2 &&& \le 11;\\
-3x_1 + 2x_2 &&& \le 10;\\
3x_1 + 4x_2 &&& \ge 20;\\
x_1, x_2 &&& \ge 0.
\end{aligned}
$$

3.6.

$$
\begin{aligned}
f &= x_1 + x_2 + 2x_3 + x_4 && \to \max;\\
x_1 + 2x_2 + 2x_3 + x_4 &&& \le 40;\\
x_1 + 3x_2 &&& \le 20;\\
x_1 + 3x_2 &&& \le 10;\\
x_3 &&& \le 10;\\
x_4 &&& \le 10;\\
x_3 + x_4 &&& \le 15;\\
x_j &&& \ge 0,\ j = \overline{1,5}.
\end{aligned}
$$

4. Розв'язати задачі п. 3, якщо $x_i$ -- цілі.

# Розділ 3. Методи оптимізації нелінійних функцій без обмежень

## 3.1. Постановка задачі

Нехай функція $f(x)$, $x=[x_1,x_2,\ldots,x_n]^T$, визначена у $n$-вимірній області $G\subset R^n$. Точкою відносного екстремуму (мінімуму) фнукції $f(x)$ називають внутрішню точку $x^*\in G$, в якій $f(x^*)\le f(x)$, тобто існує число $rho>0$ таке, що $f(x^*)\le f(x)$ за умови $\|x-x^*\|<\rho$.

Якщо незалежні змінні $x_1,x_2,\ldots,x_n$ мають додаткову залежність у вигляді умов $\varphi_i(x)=0$, $i=\overline{1,m}$, точку $x^*$) називають точкою умовного екстремуму функції $f(x)$.

Розглянемо задачу пошуку точок відносного мінімуму. Нехай $M(x)$ -- точка відносного мінімуму функції і у деякому її околі існують похідні $\partial f(x)/\partial x_i$; $i=\overline{1,n}$, і нехай $M_0(x^*)$ -- деяка точка, що досить близька до точки $M_0(x)$.

Позначимо $x_i=x_i(t)$ -- параметричне рівняння траєкторії, що проходить у момент часу $t=0$ через точку $x^0$, вздовж якої за збільшення $t$ функція $f(x)$ спадає і при цьому найшвидше. Тоді, оскільки вектор

$$\text{grad}f=\left[\frac{\partial f}{\partial x_1},\frac{\partial f}{\partial x_2},\ldots,\frac{\partial f}{\partial x_n}\right]^T$$

спрямований у бік найсильнішого зростання функції $f(x)$, можна записати

$$\frac{dx}{dt}=-\text{grad}f(x). \qquad (136)$$

звідси випливає, що траєкторія $x=x(t)$ найшвидшої зміни функції $f(x)$ є ортогональна до відповідних поверхонь рівня $f(x)=\text{const}$. У скалярному вигляді рівняння (1) записується як:

$$\frac{dx_1}{dt}=-\frac{\partial f}{\partial x_1},\ldots,\frac{dx_n}{dt}=-\frac{\partial f}{\partial x_n}. \qquad (137)$$

Щоб відшукати частинний розв'язок цієї системи рівнянь, доповнимо її початковими умовами

$$x_i(0)=x_{i0},\ i=\overline{1,n}. \qquad (138)$$

Задача (2) , (3) -- задача Коші. На практиці цю задачу розв'язують наближено, апроксимуючи $dx_i/dt$ за формулою Ейлера

$$x_i^{j+1}=x_i^j-h\frac{\partial f_j}{\partial x_i},\ i=\overline{1,n}. \qquad (139)$$

де $f_j=f(x_1^j,x_2^j,\ldots,x_n^j)$; $x_i^j=x_i(t_j)$; $t_j=jh$. Крок $h$ обирають за умови $f_{j+1}<f_j$.

Отже, якщо точка $x$ вже знайдена, після обчислення $\partial f_j/\partial x_1$, $\partial f_j/\partial x_2,\ldots$, $\partial f_j/\partial x_n$, можна відшукати точку $x^{j+1}$, для якої функція $f(x)$ матиме менше за попередню точку

значення.

Розглянемо деякі наближені методи пошуку точок екстремуму функції $f(x)$, що не потребують розв'язання системи рівнянь (2).

Як відомо, умови існування екстремуму визначаються такими теоремами.

**Теорема.** *Для того, щоб функція $n$ змінних $f(x)$, що визначена на дійсній осі, мала безумовний екстремум у точці $x$, необхідно, щоб виконувалася умова*

$$\frac{\partial f}{\partial x_j}|_{x=x^*} = 0, \quad j = \overline{1,n}. \tag{140}$$

Точка $x^*$, яка задовольняє (5), зветься стаціонарною точкою.

**Теорема.** *Для того, щоб подвійно диференційовна функція $n$ змінних $f(x)$ мала у стаціонарній точці $x$ безумовний локальний екстремум, необхідно, щоб матриця її других похідних була додатно (від'ємно) визначена.*

Перевірка знаковизначеності матриць може бути виконана, наприклад, за допомогою критерію Сільвестра, згідно з яким необхідною та достатньою умовою додатної визначеності квадратичної форми $(x, Ax)$, де $A = \{a_{ij}\}$ -- симетрична $n \times n$ матриця, є виконання нерівностей

$$a_{11} > 0, \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} > 0, \ldots, \begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \ldots & & & \\ a_{n1} & a_{n2} & \ldots & a_{nn} \end{pmatrix} > 0. \tag{141}$$

Якщо квадратична форма не змінює знаку, але дорівнює нулю за ненульових значень аргументу, визначення характеру стаціонарної точки $x$ потребує додаткового дослідження похідних вищого порядку.

Щоб знайти розв'язання задачі, побудуємо ітераційний процес

$$x^{j+1} = x^j + \alpha^j p^j; \quad j = 1, 2, \ldots, . \tag{142}$$

де $p^j$ -- вектор, що визначає напрямок руху із точки $x^j$, $\alpha^j$ -- числовий множник, величина якого визначає довжину кроку у напрямку $p^j$.

Процес (7) можна визначити, якщо будуть указані засоби побудови вектора $p^j$ та обчислені $\alpha^j$ на кожній ітерації. Щоб наблизитися до точки $x^*$, необхідно рухатися із точки $x^j$ у напрямку убування функції, тобто вектор $p^j$ має задовольняти умову

$$(f'(x^j), p^j) < 0.$$

Усі ітераційні процеси, в яких напрямок руху на кожному кроці збігається із антиградієнтом функції, мають назву *градієнтних методів* і відрізняються один від одного засобами вибору кроку. Найрозповсюдженішими є такі два засоби вибору кроку.

## 3.2. Методи одновимірного пошуку

Існує група методів пошуку екстремуму функцій, що не потребує визначення похідних функції, яка мінімізується. Ці методи належать до групи методів прямого одновимірного пошуку екстремуму функцій. Оскільки ефективність цих методів досить низька, ми

розглянемо два найпопулярніші методи: метод ``золотого перетину'' та метод Фібоначчі. Вони застосовуються для пошуку екстремуму унімодальних функцій, отобо функцій, що мають на визначеному інтервалі єдиний екстремум.

### 3.2.1. Метод ``золотого перетину''

``Золотим перетином'' відрізка зветься поділ його на дві нерівні частини таким чином, щоб відношення довжини усього відрізку до довжини більшої частини дорівнювало відношенню довжини більшої частини до довжини меншої частини відрізку. Для поділу відрізка $[a,b]$ використовуються два дроби Фібоначчі

$$F_1 = \frac{3-\sqrt{5}}{2} \approx 0,382;\ F_2 = \frac{\sqrt{5}-1}{2} \approx 0,618. \qquad (143)$$

Ці числа задовольняють рівностям: $F_1 = F_2^2$ ; $F_1 + F_2 = 1$ . Маємо $u_1 = a + F_1(b-a) \approx a + 0,382(b-a)$ ; $u_2 = a + F_2(b-a) \approx a + 0,618(b-a)$ ; $a < u_1 < u_2 < b$ . Своєю чергою числа $u_1$ , $u_2$ також продукують ``золотий перетин'' відповідно $[a,u_2]$ та $[u_1,b]$.

Виходячи з цього, алгоритм методу можна побудувати таким чином. Нехай $a_1 = a$ , $b-1 = b$ . На відрізку $[a_1,b_1]$ візьмемо точки $u_1,u_2$ , що продукують ``золотий перетин'', та обчислимо значення функції $f(u_1)$ , $f(u_2)$ . Якщо $f(u_1) \le f(u_2)$ , $a_2 = a_1$ , $b_2 = u_1$ ; $\bar{u}_2 = u_1$ . Якщо $f(u_1) > f(u_2)$ , $a_2 = u_1$ , $b_2 = b_1$ ; $\bar{u}_2 = u_2$ ; $b_2 - a_2 = F_2(b-a)$ та, що важливо, точка $\bar{u}_2 \in [a_2,b_2]$ , в якій $f(\bar{u}_2) = \min\{f(u_1), f(u_2)\}$ виконує ``золотий перетин'' $[a_2,b_2]$.

Якщо кількість ітерацій не обмежена, процес триває доти, доки не справдиться умова

$$b_n - a_n < \varepsilon. \qquad (144)$$

За наявності більше одного мінімуму функції $f(x)$ на відрізку $[a,b]$ отриманий розв'язок може опинитися далеко від глобального мінімуму функції. Щоб запобігти цьому, можна запропонувати таку модифікацію алгоритму.

Нехай нам відома початкова точка $x^{(0)}$. Початковий відрізок $\Delta$, в якому міститься мінімум функції $f(x)$, можна визначити, наприклад, за допомогою послідовної серії кількох зростаючих кроків за незалежною змінною. Позначимо три послідовні значення $x$ через $x_3^{(0)}$ (остання точка), $x_2^{(0)}$ та $x_1^{(0)}$, де $f(x_3^{(0)}) \ge f(x_2^{(0)})$. Нехай $\Delta^{(0)} = x_3^{(0)} - x_1^{(0)}$. Якщо уже виконано $k$ кроків, $(k+1)$-й інтервал визначатимемо так:

$$u_1^{(k)} = x_1^{(k)} + F_1\Delta^{(k)}, \qquad (145)$$

$$u_2^{(k)} = x_1^{(k)} + F_2\Delta^{(k)} = x_3^{(k)} - F_1\Delta^{(k)}. \qquad (146)$$

Далі, якщо

$$f(u_1^{(k)}) < f(u_2^{(k)}),\ \Delta^{(k+1)} u_2^{(k)} - x_1^{(k)},\ x_1^{(k+1)} = x_1^{(k)};\ x_3^{(k+1)} = u_2^{(k)}. \qquad (147)$$

$$f(u_1^{(k)}) > f(u_2^{(k)}),\ \Delta^{(k+1)} x_3^{(k)} - u_1^{(k)},\ x_1^{(k+1)} = u_1^{(k)};\ x_3^{(k+1)} = x_3^{(k)}. \qquad (148)$$

Визначення двох нових точок замість однієї у попередньому алгоритмі гарантує визначення мінімуму функції та підвищує швидкість збіжності.

### 3.2.2. Метод Фібоначчі

Метод Фібоначчі також належить до симетричних методів. Він грунтується на використанні чисел Фібоначчі, які визначаються співвідношеннями:

$$F_{n+1} = F_n + F_{n-1}; n = 2,3,\ldots; \; F_1 = F_2 = 1. \tag{149}$$

Числа Фібоначчі визначаються формулою

$$F_n = \left[ \left( \frac{1+\sqrt{5}}{2} \right)^n - \left( \frac{1-\sqrt{5}}{2} \right)^n \right] \frac{1}{\sqrt{5}}; n = 1,2,\ldots \tag{150}$$

Визначається точка $u_1^{(n)} = a + (b-a)F_{n-1}/F_{n+1}$ або симетрична до неї точка $u_2^{(n)} = a + (b-a)F_n/F_{n+1}$.

$$\Delta_k = b_k - a_k = (b-a)\frac{F_{n-k+2}}{F_{n+1}}. \tag{151}$$

Точка $u_k \in (a_k, b_k)$, в якій обчислено значення $fu_k) = \min\limits_{1 \le i \le k} f(u_k)$, збігається з однією із точок

$$u'_k = a_k + (b_k - a_k)\frac{F_{n-k}}{F_{n-k+2}} = a_k + (b-a)\frac{F_{n-k}}{F_{n+1}}, \tag{152}$$

$$u''_k = a_k + (b_k - a_k)\frac{F_{n-k+1}}{F_{n-k+2}} = a_k + (b-a)\frac{F_{n-k+1}}{F_{n+1}} = a_k + b_k - u'_k, \tag{153}$$

що містяться на відрізку $[a_k, b_k]$ симетрично відосно його середини.

Оскільки застосування вищевикладених методів не надає можливості відшукувати точки мінімуму із достатньою точністю (досягнення високої точності результату пов'язане із значною кількістю ітерацій). ми будемо застосовувати ці методи для визначення допустимих точок при розв'язанні задач із обмеженнями, оскільки при цьому досить знайти множину точок, що належать допустимій області.

### 3.2.3. Метод сполучених напрямків Пауела

Із алгоритмів прямого пошуку метод сполучених напрямків Пауела є найефективніший. Він орієнтований на розв'язання задач із квадратичними функціями мети. Задачі із квадратичними функціями мети займають важливе місце у теорії оптимізації з двох причин.

1. Квадратична функція є найпростіший тип нелінійних функцій, для яких може бути сформульовано задачу безумовної оптимізації.

2. В околі точки оптимуму будь-яку нелінійну функцію можна апроксимувати квадратичною формою, отже, розв'язання задач із квадратичною функцією надає можливість отримати певне уявлення про збіжність алгоритму у випадку мінімізації функцій загального вигляду.

Основна ідея алгоритму полягає у тому, що якщо квадратична функція $n$ змінних приведена до вигляду суми повних квадратів, її оптимум можна відшукати як наслідок реалізації $n$ одновимірних пошуків за перетвореними координатними напрямками.

Процедура перетворення квадратичної функції

$$q(x) = a + b^T x + 1/2 x^T C x \tag{154}$$

до вигляду суми повних квадратів еквівалентна пошуку такої матриці перетворення $T$, яка приводить матрицю квадратичної форми до діагонального вигляду. Тобто, $q(x0 = x^T C x$

перетворенням $x = Tz$ зводиться до вигляду

$$q(x) = z^T T^T C z = z^T D z \qquad (155)$$

де $D$ -- діагональна матриця. Нехай $t_j$ -- $j$-й стовпчик матриці $T$. Тоді перетворення $x^T Tz$ надає можливість записати кожний вектор $x$ у вигляді лінійної комбінації вектор-стовпців $t_j$ :

$$x = Tz = t_1 z_1 + t_2 z_2 + \cdots + t_n z_n. \qquad (156)$$

Система векторів $t_j$ відповідає головним осям квадратичної форми що розглядається, оскільки матриця квадратичної форми зводиться до діагонального вигляду.

Отже, одновимірний пошук оптимуму у просторі перетворених змінних $x$ еквівалентний пошуку вздовж кожної із головних осей квадратичної форми. Оскільки напрямки головних осей визначаються векторами $t_j$, одновимірний пошук виконується у напрямку, що заданий цими векторами.

*Приклад.* Розглянемо функцію

$$f(x) = 4x_1^2 + 3x_2^2 - 4x_1 x_2 + x_2$$

і перетворення

$$x_1 = z_1 + 1/2 x_2;\ \ x_2 = z_2;\ x = \begin{bmatrix} 1 & 1/2 \\ 0 & 1 \end{bmatrix} z.$$

Маємо

$$f(z) = 4z_1^2 + 2z_2^2 + z_1 + 1/2 z_2.$$

Для початкової точки $x^0 = [0, 0]^T$ отримаємо точку $[-3/16, -1/8]^T$ внаслідок проведення двох послідовних пошуків у напрямках $t_1 = [1, 0]^T$ та $t_1 = [1/2, 1]^T$. Пошук у напрямку $t_1$ за формулою $x^{(1)} = x^{(0)} + \lambda t_1$ дає $\lambda = -1/8$ та точку $x^{(1)} = [-1/8, 0]^T$. Із точки $x^{(1)}$ виконується пошук у напрямку $t_2$. Отримуємо $\lambda = -1/8$ і розв'язання $x^{(2)} = [-3/16, -1/8]^T$.

Отже, якщо система векторів $t_j$, $j = \overline{1, n}$, або система сполучених напрямків, побудована, точку оптимуму квадратичної функції можна знайти внаслідок реалізації точно $n$ одновимірних пошукіів, що проводяться уздовж кожного із $n$ напрямків $t_j$. Якщо матриця $C$ відома, матрицю $T$ перетворення можна отримати за допомогою методу *Гауса--Жордана*, який дозволяє представити матрицю $C$ у вигляді добутку

$$C = P^T D P, \qquad (157)$$

$$(P^{-1})^T C P^{-1} = D \text{òà} T = P^{-1}. \qquad (158)$$

Нехай маємо квадратичну функцію $q(x)$, дві довільні точки $x^{(1)}$ і $x^{(2)}$, що не збігаються, а також напрямок $d$. Якщо точка $y^{(1)}$ мінімізує функцію $q(x^{(1)} + \lambda d)$, а точка $y^{(2)}$ мініміззує функцію $q(x^{(2)} + \lambda d)$, напрямок $y^{(2)} - y^{(1)}$ є сполучений із $d$.

**Сполучені напрямки**

Нехай $C$ -- симетрична матриця порядку $m \times n$; напрямки $s^{(1)}, s^{(2)}, \ldots, s^{(r)}$, $r \le n$, мають назву $C$ -сполучених, якщо ці напрямки лінійно незалежні та

$$s^{(i)} C s^{(i)} = 0 \text{äëÿ} \text{ó} \tilde{n}^3 \text{ð} i \ne j. \qquad (159)$$

Знову розглянемо квадратичну функцію загального вигляду

$$q(x) = a + b^T x + 1/2 x^T C x.$$

Точки прямої, що виходить із $x^{(1)}$ у напрямку $d$, задаються формулою

$$x = x^{(1)} + \lambda d.$$

Маємо

$$\frac{\partial q}{\partial \lambda} = \frac{\partial q}{\partial x} \frac{\partial x}{\partial \lambda} (b^T + x^T C) d. \qquad (160)$$

Тоді, якщо рухатися із точки $x^{(1)}$, мінімум досягається у точці $y^{(1)}$:

$$[(y^{(1)})^T C + b^T] d = 0.. \qquad (161)$$

Аналогічно, якщо рухатися із точки $x^{(2)}$, мінімум досягається у точці $y^{(2)}$:

$$[(y^{(2)})^T C + b^T] d = 0.. \qquad (162)$$

Звідси маємо

$$(y^{(2)} - y^{(1)})^T C d = 0.$$

Отже, напрямки $y^{(2)} - y^{(1)}$ та $d$ виявляються $C$ -сполученими.

Цим доведена властивість *паралельного підпростору* для квадратичної функції.

**Реалізація сполучених напрямків Пауела**

Алгоритм містить такі кроки:

1. Задати початкову точку $x^{(0)}$ та систему $n$ лінійно незалежних напрямків; можливий випадок, коли $s^{(i)} = e^{(i)}$, $i = \overline{1, n}$.

2. Мінімізувати $F(x)$ при послідовному руху за $(N+1)$ напрякмами, при цьому точка мінімуму, яку уже отримано, обирається як вхідна, а напрямок $s^{(n)}$ використовується як на першому, так і на останньому пошуках.

3. Визначити новий сполучений напрямок за допомогою властивості паралельного підпростору.

4. Замінити $s^{(1)}$ на $s^{(2)}$ тощо. Замістити $s^{(n)}$ сполученим напрямком. Перейти до кроку 2.

*Приклад 2.* Знайти точку мінімуму функції

$$f(x) = 2x_1^3 + 4x_1 x_2^3 - 10 x_1 x_2 + x_2^2,$$

якщо задано початкову точку $x^{(0)} = [5, 2]^T$, в якій $f = 314$.

$$1. s^{(1)} = [1, 0]^T = e^{(1)}, s^{(2)} = [0, 1]^T = e^{(2)}.$$

2. а). Знайдемо таке значення $\lambda$, щоб

$$f(x^{(0)} + \lambda s^{(2)}) \to \min.$$

Отримуємо $\lambda^* = -0,81$. Звідси

$$x^{(1)} = [5, 2]^T - 0,81[0,1]^T = [5,1,19]^T,\ f(x^{(1)}) = 250.$$

b) Знайдемо таке значення $\lambda$, щоб

$$f(x^{(1)} + \lambda s^{(1)}) \to \min_{\lambda}.$$

$$\lambda^* = -3,26;\ x^{(2)} = [1,74,1,19]^T,\ f(x^{(2)}) = 1,10.$$

c) Знайдемо таке значення $\lambda$, щоб

$$f(x^{(2)} + \lambda s^{(2)}) \to \min_{\lambda}.$$

$$\lambda^* = -0,098;\ x^{(3)} = [1,74,1,092]^T,\ f(x^{(3)}) = 0,72.$$

3. Покладемо $s^{(3)} = x^{(3)} - x^{(1)} = [-3,26, -0,098]^T$. після нормування отримаємо

$$s^{(3)} = \frac{s^{(3)}}{\| s^{(3)} \|} = [-0,99955, -0,03]^T.$$

Покладемо $s^{(1)} = s^{(2)}$, $s^{(2)} = s^{(3)}$ і перейдемо до кроку 2 алгоритму.

4. Знайдемо таке значення $\lambda$, щоб

$$f(x^{(3)} + \lambda s^{(2)}) \to \min_{\lambda}.$$

$$\lambda^* = 0,734,\ x^{(4)} = [1,006,1,070]^T;\ f(x^{(4)}) = -2,86.$$

## 3.3. Градієнтні методи

Якщо у (7) прийняти $p^j = -f'(x^j)$, отримаємо ітераційний процес

$$x^{j+1} = x^j - \alpha^j f'(x^j),\ \alpha^j > 0,\ \ j = 0,1,2,\ldots. \qquad (163)$$

У покоординатній формі процес (8) записується у вигляді

$$x_i^{j+1} = x_i^j - \alpha^j \frac{\partial f(x^j)}{\partial x_i},\ \ i = \overline{1,n}.$$

Параметр $\alpha^j$ можна обирати таким чином.

$$x^1 == x^0 - \alpha^0 f'(x^0).$$

Потім перевіряємо виконання нерівності

$$f(x^1) - f(x^0) \le -\varepsilon\alpha \| f'(x^0) \|^2, \qquad (164)$$

де $0 < \varepsilon < 1$ -- стала, що вибирається довільно, й яка є однією й тією самою за усіх $j = 0,1,\ldots$ Якщо нерівність (29) виконується, $\alpha^j = \alpha^0$. Якщо ні -- ділимо $\alpha$ поки ця нерівність не справдиться.

Обґрунтованість такого вибору $\alpha$ забезпечується теоремою:

**Теорема.** *Якщо функція* $f(x)$ *обмежена знизу,* grad$f(x)$ *задовольняє умову Ліпшиця*

$$\| f''(x) - f''(y) \| \le q \| x - y \| \qquad (165)$$

*за будь-яких* $x, y \in R^n$, *а вибір значення* $\alpha^j$ *виконується за вказаним способом, для процесу (28) буде виконуватися* $\| f'(x) \| \to 0$, $j \to \infty$, *якою б не була початкова точка.*

Умова (30) виконується за $\alpha \le q(1-\varepsilon)/q$. Отже, коли (30) виконується, то якщо визначити $q$ із (30), можна вибирати $\alpha = (1-\varepsilon)/q$.

Такий спосіб вибору кроку $\alpha$ характеризує градієнтний метод спуску до екстремуму.

### 3.3.1. Метод найскорішого спуску

Якщо прийняти $p^j = -f'(x^j)$, отримаємо ітераційний процес

$$x^{j+1} = x^j - \alpha^j f'(x^j), \quad \alpha^j > 0; \quad j = 0,1,2,\ldots. \tag{166}$$

У покоординатній формі процес ((46)) записується у вигляді

$$x_i^{j+1} = x_i^j - \alpha^j \frac{\partial f(x^j)}{\partial x_i}, \quad i = \overline{1,n}.$$

Параметр $\alpha^j$ можна вибирати наступним чином.

$$x^1 = x^0 - \alpha^0 f'(x^0).$$

Потім перевіряємо виконання нерівності

$$f(x^1) - f(x^0) \le -\varepsilon\alpha \| f'(x^0) \|^2, \tag{167}$$

де $0 < \varepsilon < 1$ -- стала, що вибирається довільно, яка є однією й тією самою за усіх $j = 0,1,\ldots$. Якшо нерівність (32) виконується, $\alpha^j = \alpha^0$. Якщо ні, ділимо $\alpha$ доти, доки вона не виконається.

Обгрунтованість такого засобу вибору $\alpha$ забезпечується теоремою:

**Теорема.** *Якшо функція* $f(x)$ *обмежена знизу,* $gradf(x)$ *задовольняє умову Ліпшиця*

$$\| f''(x) - f''(y) \| \le q \| x - y \| \tag{168}$$

за будь-яких $x, y \in R_n$, а вибір значення $\alpha^j$ виконується за вказаним засобом, то для процесу (28) буде виконуватись $\| f'(x \| j \to \infty \to 0$, якою б не була початкова точка.

Умова (30) виконується при $\alpha \le q(1-\varepsilon)/q$. Таким чином, якщо (30) виконується, то якщо визначити $q$ із (30), можна вибирати $\alpha = (1-\varepsilon)/q$. Такий засіб вибору кроку $\alpha$ характеризує градієнтний метод спуску до екстремуму.

При переході з точки $x^k$ у точку $x^{k+1}$ мінімізується по $\alpha$ функція $f\left(x^k - \alpha f'(x^k)\right)$, тобто

$$\begin{aligned} \varphi(\alpha) &= f\left(x^k - \alpha f'(x^k)\right) = \min_{\alpha \ge 0} f\left(x^k - \alpha f'(x^k)\right), \\ \frac{d\varphi(\alpha^k)}{d\alpha} &= -f'(x^k) f'(x^k) = 0. \end{aligned} \tag{169}$$

З цієї причини градієнтний метод пошуку екстремуму функції зветься *методом найскорішого спуску*. Напрямки спуску на двох послідовних ітераціях взаємно ортогональні.

*Визначення.* Сильно випуклою зветься подвійно неперервно диференційовна функція, матриця других похідних якої за будь-яких $x, y \in E_n$ задовольняє умову

$$m \| y\} \|^2 \le \left( f''(y), y \right) \le M \| y \|^2,$$

де $M > m > 0$ -- деякі числа.

**Теорема.** *Нехай* $f(x)$ -- *сильно випукла функція, а послідовність* $\{x^*\}$ *будується за методом (28) з вибором кроку за схемою (31). Тоді послідовність* $\{x^*\}$ *збігається до точки мінімуму* $x^*$ *зі швидкістю геометричної прогресії із знаменником*

$$q = \frac{M-m}{M+m},$$

тобто за досить великих $k$ виконується нерівність

$$\| x^{k+1} - x^k \| \le \frac{M-m}{M+m} \| x^k - x^* \|. \qquad (170)$$

Числа $M$ і $m$ можна обирати як максимальне і мінімальне власні значення матриці $f''(x^*)$. Якщо ці числа мало відрізняються одне від одного, $q \ll 1$, матриця других похідних добре обумовлена і збіжність методу досить висока. Якщо ж $m/M \ll 1$, $q$ є близьким до одиниці і градієнтні методи починають збігатися повільніше. Якщо числа $M$ і $m$ суттєво відрізняються, топологія поверхні рівня $f(x) = \text{const}$ має яружну структуру (а в задачах максимізації функції є круті гребні). У цьому випадку траєкторія градієнтного методу характеризується досить швидким спуском на ``дно'' яру, а потім повільним рухом у точку екстремуму.

Одним з можливих засобів виходу із цієї ситуації є зміна масштабу $\mu_i$ незалежних змінних функції мети. Для квадратичних функцій

$$f(x) = \sum_{i=1}^{n} \alpha_i x_i^2$$

можна обирати $\mu_i = 1/\sqrt{\alpha_i}$ .

У випадку неквадратичної функції загального вигляду вибирають

$$\mu_i = \sqrt{\frac{\partial^2 f(x)}{\partial x_i^2}} \qquad (171)$$

Таке масштабування надає можливість зменшити властивість витягнутості функції мети. Гарантовано ж виправити топологію функцій мети можна застосуванням перетворенням вигляду

$$y = \sqrt{f''(x)}\,x.$$

Таке перетворення призводить до методу Ньютона. Масштабування змінних у загальному випадку призводить до ітераційного процесу

$$x^{(k+1)} = x^{(k)} - \alpha^{(k)} B^{(k)} f'(x^{(k)}). \qquad (172)$$

У цьому рівнянні елементами матриці $B$ є масштабуючі множники.

Методи вигляду (37) часто називають *релаксаційними.* Метод найскорішого спуску зазвичай ще має назву *методу Гауса*.

*Приклад.* Знайти мінімум функції

$$f(x,y) = 2x^2 + xy + y^2 + 2x + y + 1$$

за методом найскорішого спуску.

Виберемо початкову точку спуску до екстремуму:

$$x^0 = 0,\ y^0 = -1;\ \alpha = 0,5.$$

Обчислимо у цій точці значення функції та її похідних.

$$f(x^0, y^0) = x(2x + y + 2) + y(y + 1) + 1 = 1;$$

$$\frac{\partial f}{\partial x} = 4x + y + 2 = 1;\ \frac{\partial f}{\partial y} = 2y + x + 1 = -1.$$

Обчислюємо $x^1 = y^0 - \alpha f(x^0, y^0) = 0 - 0,5 \cdot 1 = -0,5;$

$$f(x^1, y^1) = (-0,5) \cdot 0,5 + (-0,5) \cdot 0,5 + 1 = 0,5.$$

Перевіряємо виконання нерівності ((45)) при $\varepsilon = 0,5$ :

$$f(x^1, y^1) - f(x^0, y^0) = 0,5 - 1 = -0,5 \le 0,5 \cdot 0,5 \| f'(x^0, y^)) = -0,25.$$

Для вибору кроку $\alpha$ на наступному наближенні знайдемо

$$\frac{\partial \Phi(\alpha)}{\partial \alpha} = \frac{\partial f(x^1 - \alpha f'(x^1))}{\partial \alpha} = \frac{\partial \alpha^2 - 0,25\alpha + 0,5}{\partial \alpha} = 2\alpha - 0,25 = 0.$$

Значить, $\alpha^1 = 0,125$ .

Повторюємо обчислення (друга ітерація):

$$x^2 = x^1 - \alpha f(x^1, y^1) = -0,5 - 0,125 \cdot (-0,25) = -0,438;$$

$$y^2 = y^1 - \alpha f(x^1, y^1) = -0,5 - 0,125 \cdot (-0,25) = -0,438;$$

$$\frac{\partial \Phi(\alpha)}{\partial \alpha} = 0,46\alpha - 0,052 \to \alpha^{(2)} = 0,114.$$

Обчислюємо $x^3 = -0,415$ , $y^3 = -0,403$ ; $f(x^3, y^3) = 0,834$ .

Оскільки $F(x^3, y^3) > f(x^2, y^2)$ , ітераційний процес зупиняємо і розв'язання задачі -- точка мінімуму дорівнює $x^* = -0,438$ , $y^* = -0,438$ .

## 3.4. Метод сполучених градієнтів

Метод *сполучених градієнтів*, як й методи градієнтного спуску, є метод першого порядку, але вигідно відрізняється більш високою швидкістю збіжності, яка за деяких припущень відносно функції, що мінімізується, наближається до швидкості збіжності методу Ньютона.

Додатно визначена квадратична форма $n$ змінних мінімізується за методом сполучених градієнтів за $n$ або менше кроків. Оскільки будь-яка гладка функція в околі точки свого мінімуму гарно апроксимується квадратичною функцією, метод сполучених градієнтів успішно застосовується й для мінімізації неквадратичних функцій.

*Визначення.* Два вектори $x$ та $y$ у просторі $E_n$ називають $H$ - *сполученими*, або $H$ - *ортогональними*, якщо

$$(x, Hy) = 0.$$

(Якщо $H = E_n$ , вектори $x$ та $y$ ортогональні).

**Приклад**

Розглянемо спочатку задачу мінімізації функції

$$f(x) = (x_1)^2 + 4(x_2)^2 - 1.$$

Нехай $x^0$ -- довільна початкова точка, а $s^)$ -- довільний напрямок руху до точки мінімуму. Тоді наступну точку $x^1$ визначимо формулою

$$x^1 = x^0 + \lambda s^0.$$

У цій формулі довжину кроку обчислимо за умови мінімуму функції $f(x)$ по $\lambda$ у напрямку руху:

$$\frac{df(x^0 + \lambda^0 s^0)}{d\lambda^0} = 0.$$

Визначимо $x^0 = [1,1]^T$; $s^0 = [1,2]^T$. Тоді маємо $x_1^1 = 1 + \lambda^0$; $x_2^1 = 1 + 2\lambda^0$. Для обчислення довжини кроку маємо рівняння

$$\frac{df}{d\lambda^0} = 2(1+\lambda^0) + 16(1+2\lambda^0) = 0.$$

Звідси $\lambda^0 = -9/17$; $x_1^1 = 6/17$; $x_2^1 = -1/17$. Візьмемо тепер напрямок руху $s^1$ з точки $x^1$ сполученим до $s^0$ відносно матриці других похідних функції мети, тобто за умови

$$\left(s^1, f''(x)s^0\right) = 0.$$

У даному випадку матриця других похідних стала й дорівнює

$$f'' = \begin{bmatrix} 2 & 0 \\ 0 & 8 \end{bmatrix},$$

а напрямок мінімізації $s^1$ має вигляд

$$s_1^1 = 1; \quad s_2^1 = -1/8; \quad x_1^2 = x_1^1 + \lambda^1 s_1^1 = 8/17 + \lambda^1;$$
$$x_2^2 = x_2^1 + \lambda^1 s_2^1 = 8/17 - 1/8\lambda^1.$$

Знову визначимо $\lambda^1$ шляхом мінімізації функції $fx)$ за обраним напрямком

$$\frac{df}{d\lambda^1} = (x^1 + \lambda^1 s^1) = 2\left(\frac{8}{17} + \lambda^1\right) + \left(\frac{1}{17} + \frac{1}{8}\lambda^1\right) = 0. \quad \lambda^1 = -8/17.$$

Тоді

$$x_1^2 = \frac{8}{17} - \frac{8}{17} = 0; \quad x_2^2 = \frac{1}{17} - \frac{1}{8}\left(-\frac{8}{17}\right) = 0.$$

Тобто, квадратична функція двох змінних мінімізується за два кроки, по одному у кожному із сполучених напрямків.

Розглянемо тепер квадратичну функцію $n$ змінних

$$f(x) = a + (x,b) + 1/2(x, Hx), \tag{173}$$

де $H$ -- додатно визначена $(n \times n)$ матриця. Для мінімізації функції $f(x)$ побудуємо ітераційний процес

$$x^{k+1} = x^k + \lambda^k s^k, \tag{174}$$

де $s^k$ -- напрямок спуску на $k$-му кроці, а величина кроку визначається за умови мінімуму функції $f(x)$ у напрямку руху

$$\lambda^k = \frac{\left(f'(x^k), s^k\right)}{\left(s^k, Hs^k\right)}. \tag{175}$$

Обираємо початкове наближення $x^0$. Якщо застосувати послідовно формули (39),(40), то на $n$-му кроці ітераційного процесу отримаємо

$$x^n = x^0 + \sum_{k=0}^{n-1} \lambda^k s^k = x^0 - \sum_{k=0}^{n-1} \frac{\left(f'(x^k), s^k\right)}{\left(s^k, Hs^k\right)} s^k. \tag{176}$$

При цьому $s^0 = -f(x^0)$.

Зауважимо, що мінімум функції (38) досягається у точці $x$, в якій

$$f'(x) = b + Hx = 0.$$

Тоді маємо

$$x = -H^{-1}b. \tag{177}$$

Вектор $s^k$ визначається як розв'язок системи рівнянь

$$\left(s^k, Hs^{k-1}\right) = \left(s^k, Hs^{k-2}\right) = \cdots = \left(s^k, Hs^0\right) = 0. \tag{178}$$

Отже, щоб скористатися викладеною схемою мінімізації функції $f(x)$, треба мати $n$ взаємно сполучених напрямків $s^0, s^1, \ldots, s^n$.

## 3.5. Метод Флетчера--Рівса

*Метод Флетчера-Рівса* використовує послідовність напрямків пошуку, кожний з яких є лінійною комбінацією антиградієнту у поточній точці та попереднього напрямку пошуку. Розглянемо знову квадратичну функцію

$$f(x) = a + (x, b) + 1/2(x, Hx), \tag{179}$$

При мінімізації цієї функції за методом Флетчера-Рівса вектори $x^k$ обчислюються за формулами

$$s^0 = -f'(x^0), \tag{180}$$

$$s^k = -f'(x^k) + \beta^{k-1} s^{k-1}; \quad k \geq 1. \tag{181}$$

Значення $\beta^{k-1}$ вибираються так, шоб вектори $s^k, s^{k-1}$ були $H$-сполученими:

$$\left(s^k, Hsk - 1\right) = -\left(f'(x^k), Hs^{k-1}\right).$$

Звідси отримуємо

$$\beta^{k-1} = \frac{\left(f'(x^k), Hs^{k-1}\right)}{\left(s^{k-1}, Hs^{k-1}\right)}; \tag{182}$$

$$\lambda^k = \frac{\left(f'(x^k), s^k\right)}{\left(s^k, Hs^k\right)}; \tag{183}$$

$$x^{k+1} = \lambda^k s^k. \tag{184}$$

Запропонована процедура мінімізації функції $f(x)$ має вигляд.

У визначеній початковій точці $x^0$ обчислюється антиградієнт $s^0 = -f'(x^0)$ .

Виконується одновимірна мінімізація у цьому напрямку й визначається точка $x^1$. У цій точці знову обчислюється антиградієнт $-f'(x^1)$. Оскільки ця точка доставляє мінімум функції $f(x)$ вздовж напрямку антиградієнта $s^0 = -f'(x^0)$, вектор $f'(x^1)$ є ортогональним до $f'(x^0)$. Далі за формулою (46) обчислюється вектор $s^1$, який буде $H$-сполученим до вектора $s^0$, й визначається мінімум функції вздовж напрямку $s^1$ тощо.

**Теорема.** *Вектори* $s^0, s^1, \ldots s^k; k < n,$ *у методі Флетчера--Рівса є* $H$*-сполученими, а градієнти* $f'(x^0), \ldots, f'(x^k)$ *-- взаємно ортогональними. Мають місце формули*

$$\beta^{k-1} = \frac{\left(f'(x^k), Hs^{k-1}\right)}{\left(s^{k-1}, Hs^{k-1}\right)}; \tag{185}$$

$$f'(x^{k+1}) = f(x^k) + \lambda^k Hs^k. \tag{186}$$

### 3.5.1. Реалізація методу Флетчера--Рівса

Для реалізації методів сполученого градієнту та Флетчера--Рівса необхідно:

• Для вибраної початкової точки $x^0$ обчислити $s^0 = -f'(x^0)$.

• На $k$-му кроці розв'язати задачу мінімізації по $\lambda \geq 0$ функції $f(x^k + \lambda^k s^k)$ відповідно до формул (32)--(33) та визначити точку

$$x^{k+1} = x^k + \lambda^k s^k.$$

• Якщо $f(x^{k+1}) = 0$, ця точка є розв'язком задачі, якщо ні -- визначити

$$s^{k+1} = -f'(x^{k+1}) + \frac{\left(f'(x^{k+1}), Hs^k\right)}{\left(s^k, Hs^k\right)} s^k. \tag{187}$$

та обчислити наступну ітерацію.

## 3.6. Метод Ньютона

Нехай $\varphi(x)$ є $n$-вимірна векторна функція векторного аргументу $x$ тієї самої розмірності. Для розв'язку системи рівнянь

$$\varphi(x) = 0$$

розкладемо покоординатно функцію $\varphi(x)$ у ряд Тейлора. Обмежившись лінійним наближенням, отримимо систему рівнянь

$$\varphi_i(x_1^k, x_2^k, \ldots, x_n^k) + \sum_{j=1}^{n} (x_j - x_j^k) \frac{\partial \varphi_i(x^k)}{\partial x_j} = 0.$$

Її розв'язок має вигляд

$$x^{k+1} = x^k - \left[\varphi'(x^k)\right]^{-1} \varphi(x^k), \tag{188}$$

де $\varphi'(x^k) = \left\{\partial \varphi_i(x^k)/\partial x_j\right\}$ -- квадратна $n \times n$ матриця.

Розглянемо тепер випадок, коли вектор-функція $\varphi(x)$ є градієнт деякої скалярної функції $f(x)$, тобто

$$\varphi(x) = f'(x) + \text{grad} f(x). \qquad (189)$$

Якщо прирівняти її до нуля, отримимо систему рівнянь, що визначає координати стаціонарних точок функції $f(x)$. Формула Ньютона для пошуку розв'язку цієї системи має такий вигляд

$$x^{k+1} = x^k - \left[ f''(x) \right]^{-1} f'(x^k). \qquad (190)$$

Ітераційний процес (44) будує послідовність точок $\{x^k\}$, яка за деяких припущень відносно функції, що мінімізується, збігається у деякій стаціонарній точці $x^*$ функції $f(x)$, тобто до точки $x^*$, в якій $f'(x^*) = 0$.

Якщо матриця других похідних $f''(x)$ (матриця Гессе) позитивно визначена, ця точка буде точкою суворого локального мінімуму функції $f(x)$. Припустимо, що матриця Гессе задовольняє умову

$$m \| y \|^2 < \left( f''(x) y, y \right) < M \| y \|^2, \quad M > m > 0,$$

для будь-яких $x, y \in E_n$.

Апроксимуємо функцію $f(x)$ в околі точки $x^k$ квадратичною формою

$$f'(x) = f(x^k) + f'(x^k)(x - x^k) + 1/2 \left( f''(x^k)(x - x^k), x - x^k \right). \qquad (191)$$

Ця форма має єдину точку мінімуму, що є коренем рівняння

$$f'(x) = 0 = f'(x^k) + f''(x^k)(x - x^k),$$

тобто збігається із точкою $x^{k+1}$.

Таким чином, метод Ньютона інтерпретується як послідовний пошук точок мінімуму квадратичних апроксимуючих функцій типу (44).

Розглянемо тепер метод Ньютона у двовимірному випадку й покажемо важливість додатної визначеності матриці Гессе. Квадратична апроксимація (56) для функції двох змінних виглядає так:

$$f(x) = b_0 + b_1 x_1^k + b_2 x_2^k + b_{11} (x_1^k)^2 + b_{22} (x_2^k)^2 + b_{21} x_2^k x_1^k + b_{12} x_1^k x_2^k.$$

Цю форму можна перетворити до канонічного вигляду

$$\tilde{f}(x^k) - \tilde{f}(x^*) = \bar{b}_{11} (x_1^k)^2 + \bar{b}_{22} (x_2^k)^2, \qquad (192)$$

де $x^*$ -- стаціонарна точка функції $f(x)$,

$$b_{11} = \frac{1}{2} \frac{\partial^2 f(x^k)}{\partial (x_1)^2}; \quad b_{12} = b_{21} = \frac{1}{2} \frac{\partial^2 f(x^k)}{\partial x_1 x_2}; \quad b_{22} = \frac{1}{2} \frac{\partial^2 f(x^k)}{\partial (x_2)^2}.$$

Коефіцієнти $\bar{b}_{11}, \bar{b}_{22}$ канонічного перетворення у (56) є власними значеннями матриці $1/2 f''(x^k)$. Якщо ці власні значення додатні, апроксимуюча квадратична функція має вигляд колової або еліптичної западини (параболоїд обертання). Якщо знаки $\bar{b}_{11}, \bar{b}_{22}$ протилежні, функція $f(x)$ являє собою гіперболічний параболоїд -- поверхню із сідловою точкою. Ця точка береться як наступне наближення $x^{k+1}$ у методі Ньютона, незважаючи на ту обставину, що величина $f(x^{k+1})$ може виявитися більшою за величину $f(x^k)$.

Отже, на збіжність методу Ньютона можна розраховувати тільки у тому випадку, коли матриця Гессе функції мети додатно визначена на кожній ітерації. Якщо числа $\bar{b}_{11}, \bar{b}_{22}$ мають один знак, але суттєво відрізняються за величиною, квадратична функція (56) має яр.

У загальному випадку коли функція, що мінімізується, не є квадратичною, вектор $\left[ f''(x^k) \right]^{-1} f'(x^k)$ вже не вказує у точку її мінімуму, однак має більшу складову вздовж віссі яру та значно ближче до напрямку на мінімум, ніж антиградієнт. Цим й обумовлюється більш висока збіжність методу Ньютона порівняно з градієнтним при мінімізації яружних функцій.

Головні недоліки методу Ньютона полягають у тому, що він обумовлює обчислення других похідних, а також може розбігатись, якщо функція мети не є сильно випуклою й початкове наближення обирається досить далеко від точки мінімуму.

Отже, вдалий вибір початкового наближення гарантує збіжність методу Ньютона, але у загальному випадку пошук відповідного початкового наближення є доволі не тривіальною задачею. У зв'язку з цією обставиною треба якось модифікувати формули (42), (44), щоб домогтися збіжності відповідних процесів незалежно від початкового наближення. У методі Ньютона для цього досить окрім напрямку руху вибирати ще й довжину кроку вздовж цього напрямку. Відповідні алгоритми називаються методами Ньютона із регулюванням кроку (методами Ньютона--Рафсона). Вони мають вигляд

$$x^{k+1} = x^k - \alpha^k \left[ \varphi'(x^k) \right]^{-1} \varphi(x^k); \tag{193}$$

$$x^{k+1} = x^k - \alpha^k \left[ f''(x^k) \right]^{-1} f'(x^k). \tag{194}$$

Розглянемо два засоби вибору довжини кроку $\alpha^k$. Перший з них полягає у перевірці нерівності

$$f(x^k) - \alpha^k \left[ f''(x^k) \right]^{-1} f'(x^k) \le \varepsilon \alpha^k \left( f'(x^k), p^k \right), \tag{195}$$

де $p^k = -\left[ f''(x^k) \right]^{-1} f'(x^k)$ -- напрямок спуску, $0 < \varepsilon < 1/2$ -- деяке число.

Якщо нерівність (45) виконується при $\alpha^k = 1$, крок приймається рівним одиниці, якщо ні -- крок дробиться доти, доки ця нерівність не справдиться.

Другий метод визначення кроку $\alpha^k$ полягає у мінімізації по $\alpha$ функції

$$f(x^k) - \alpha^k \left[ f''(x^k) \right]^{-1} f'(x^k)$$

у напрямку руху

$$f(x^k) - \alpha^k \left[ f''(x^k) \right]^{-1} f'(x^k) = \min_{\alpha \ge 0} f(x - \alpha, \ldots). \tag{196}$$

**Теорема.** *Нехай двічі неперервно диференційована функція є сильно випуклою й матриця її других похідних задовольняє умові Ліпшиця*

$$\| f''(x) - f''(y) \| < R_1 \| x - y \|, \quad x, y \in E_n.$$

*Тоді послідовність (51), у якій $\alpha^k$ вибирається згідно з (52) або (539), збігається незалежно від вибору почткового наближення $x^k$ до точки мінімуму $x^*$ із квадратичною швидкістю*

$$\| x^{k+1} - x^k \| \le \frac{R_1}{b} \| x^k - x^* \|^2, \tag{197}$$

де $b$ -- оцінка мінімального власного значення матриці $f'(x)$.

Розглянемо тепер випадок, коли матриця Гессе не є додатно визначена. Тоді, як відомо, послідовність точок $\{x^k\}$, що виробляється за методом Ньютона, може розбігатись. Для

забезпечення збіжності ітераційного процесу Ньютона у цьому випадку Левенберг та Марквардт запропонували додавати до матриці Гессе на кожному кроці величину $\lambda^k E$, де $\lambda^k$ -- деяке число. Відповідний алгоритм набуває вигляду

$$x^{k+1} = x^k - \left[ f''(x^k) + \lambda^k E \right]^{-1} f'(x^k). \qquad (198)$$

Величина $\lambda^k$ має задовольняти умови

$$\cos\left( \left[ f''(x^k) + \lambda^k E \right]^{-1} f'(x^k), f'(x^k) \right) > \varepsilon_1 > 0; \qquad (199)$$

$$f(x^{k+1}) - f(x^k) < -\varepsilon_2 (\left[ f''(x^k) + \lambda^k E \right]^{-1} f'(x^k), f'(x^k)). \qquad (200)$$

За такого визначення $\lambda^k$ забезпечується збіжність процесу.

В наведених вище нерівностях $\varepsilon_1, \varepsilon_2$ -- задані сталі. Перша нерівність означає, що кут між напрямком спуску та антиградієнтом у точці $x^k$ має бути гострим, а виконання другої умови гарантує суттєве убування функції на кожній ітерації.

Параметри $\lambda^k$ умовами ((71)), ((72)) не визначаються однозначно, й для їхнього вибору існують різні евристичні схеми. Якщо вибрати правильний спосіб вибору цих параметрів, метод ((198)) вдалині від точки мінімуму веде себе як градієнт, а при наближенні до точки мінімуму має переходити у звичайний метод Ньютона.

### 3.6.1. Реалізація методу Ньютона

Алгоритм, що реалізує метод Ньютона--Рафсона мінімізації функції $f(x)$, передбачає виконання наступних дій.

• Для вибраного початкового наближення $x^0$ обчислюються значення $f(x^0)$, $f'(x^0)$ та $f''(x^0)$.

• Обчислюється напрямок спуску до екстремуму

$$p^k = -\left[ f''(x^k) \right]^{-1} f'(x^k)$$

як розв'язок системи лінійних рівнянь

$$f''(x^k) p^k = f'(x^k); k = 0,1,2,\ldots.$$

• При $\alpha^0 = 1$ обчислити значення функції $f(x^k - \alpha^0 p^k)$ та скалярний добуток

$$r_1 = \left( f'(x^k), p^k \right) = \sum_{j=1}^{n} f'(x_j^k) p_j^k.$$

• Перевірити виконання нерівності (при $\varepsilon = 0,5$):

$$f(x^k - \alpha^k p^k) - f(x^k) \le \varepsilon \alpha^k r_1.$$

Якщо ця нерівність виконується, то $\alpha^1 = 1$, якщо ні, крок ділиться навпіл, доки вона не справдиться.

• Обчислити наступне наближення

$$x^{k+1} = x^k - \alpha^k \left[ f''(x^k) \right]^{-1} f'(x^k)$$

• Повторити обчислення за пп.2-5 при $k = 1,2,\ldots$ та перевірити нерівність

$$f(x^{k+1}) < f(x^k).$$

*Приклад.* Відшукати мінімуму функції

$$f(x,y) = 2 - 6x + x^2 + 4y^2 - 4x^2y + x^4$$

На першій ітерації маємо

$$\left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right] = d^1 = [-2, 0].$$

$$Ap^{(0)} = d^1 = \begin{bmatrix} 34 & -16 & 2 \\ -16 & 8 & 0 \end{bmatrix};$$

$$p^0 = [1\,2];\ x^1 = x^0 + p^0 = [3\,4].$$

На 2-й ітерації:

$$\left[\frac{\partial f}{\partial x^1}, \frac{\partial f}{\partial y^1}\right] = d^2 = [-12, 4].$$

$$Ap^{(1)} = d^1 = \begin{bmatrix} 78 & -24 & -12 \\ -24 & 8 & 4 \end{bmatrix};$$

$$p^1 = [0\,0,5];\ x^2 = x^1 + p^1 = [3\,4,5];\ \alpha = 1;$$

$$\left[\frac{\partial f}{\partial x^2}, \frac{\partial f}{\partial y^2}\right] = d^2 = [0\,0].$$

Отже, за методом Ньютона розв'язання задачі отримано за дві цтерації:

$$x^* = 3;\ y^* = 4,5.$$

## 3.7. Метод Девідона--Флетчера--Пауела

Розглянемо ще один метод розв'язку задачі безумовної оптимізації функцій, зважаючи на досить поширене його використання при розв'язку задач пошуку екстремуму функцій. Метод Девідона--Флетчера--Пауела (ДФП) належить до групи методів сполучених градієнтів, але, на відміну від методів сполучених градієнтів та Флетчера -Рівса, у цьому методі для пошуку екстремуму функцій застосовується апроксимація матриці Гессе або зворотної до неї, але використовуються тільки перші похідні.

Такі методи мають назву методів *змінної метрики*, або *квазіньютоновських*. При використанні методів змінної метрики новий вектор $x$ обчислюється за вектором попереднього кроку за допомогою рівняння

$$x^{(k+1)} = x^{(k)} + \mathbf{l}^{(k)} s^{(k)} = x^{(k)} - \mathbf{l}^{*(k)} \mathbf{h}(x^{(k)}) \nabla f(x^{(k)}), \qquad (201)$$

де матриця $\mathbf{h}(x^{(k)})$, що іноді зветься матрицею напрямку, являє собою апроксимацію матриці $H^{-1}(x)$.

Для обчислення матриці $\mathbf{h}(x^{(k)})$ необхідно точно обчислити другі частинні похідні від $f(x)$ та зворотну матрицю $H^{-1}(x)$. У методах змінної метрики використовуються співвідношення, що не потребують обчислення $H^{-1}(x)$.

У випадку функції мети у вигляді квадратичної форми маємо

$$\nabla f(x^{(k+1)}) - \nabla f(x^{(k)}) = H(x^{(k)})(x^{(k+1)} - x^{(k)}). \qquad (202)$$

Помножимо це рівняння на $H^{-1}(x)$. отримуємо

$$x^{(k+1)} - x^{(k)} = H^{-1}(x^{(k)})[\nabla f(x^{(k+1)}) - \nabla f(x^{(k)})]. \quad (203)$$

Якщо $f(x)$ квадратична, то $H(x) = H$ й рівняння (68) можна розгядати як систему $n$ лінійних рівнянь з $n$ невідомими, які треба оцінити для того, щоб апроксимувати $H^{-1}(x)$ або $H(x)$ при заданих значеннях $f(x)$, $\nabla f(x)$ та $\Delta x$ на попердніх етапах пошуку. Для розв'язку цієї системи лінійних рівнянь використовуються різні методи, кожний з яких характеризує різні методи змінної метрики. Матриця $H^{-1}(x)$ апроксимується наступнм чином:

$$H^{-1}(x^{(k+1)}) \approx \omega \mathbf{h}^{(k+1)} = \omega(\mathbf{h}x^{(k)} + \Delta \mathbf{h}x^{(k)}). \quad (204)$$

Визначенню підлягає матриця $\Delta \mathbf{h}(x^{(k)})$, а $\omega = const$ -- масштабуючий множник, що зазвичай дорівнює одиниці. Вибір $\Delta \mathbf{h}(x^{(k)})$ й визначає метод змінної метрики. Для забезпечення збіжності матриця $\omega \mathbf{h}(x^{(k+1)})$ має бути позитивно визначеною та задовольняти рівнянню ((76)) у тому випадку, коли вона замінює матрицю $H^{-1}(x)$.

На $(k+1)$-му кроці нам відомі вектори $x^{(k)}$), $\nabla f(x^{(k)})$, $\nabla f(x^{(k+1)})$ та матриця $\mathbf{h}(x^{(k)})$. Необхідно обчислити $\mathbf{h}(x^{(k+1)})$ таким чином, щоб задовольнялось співвідношення

$$\mathbf{h}(x^{(k+1)})\Delta g^{(k)} = \frac{1}{\omega}\Delta x^{(k)}. \quad (205)$$

Позначимо $\Delta \mathbf{h}(x^{(k)}) = \mathbf{h}(x^{(k+1)}) - \mathbf{h}(x^{(k)})$. Тоді рівняння

$$\Delta \mathbf{h}(x^{(k)})\Delta g^{(k)} = \frac{1}{\omega}\Delta x^{(k)} - \mathbf{h}(x^{(k)})\Delta g^{(k)} \quad (206)$$

треба розв'язати відносно $\Delta \mathbf{h}(x^{(k)})$. Останнє рівняння має розв'язок

$$\Delta \mathbf{h}(x^{(k)})\Delta g^{(k)} = \frac{1}{\omega}\frac{\Delta x^{(k)} y^T}{y^T \Delta g^{(k)}} - \frac{\mathbf{h}(x^{(k)})\Delta g^{(k)} z^T}{z^T \Delta g^{(k)}}, \quad (207)$$

де $y$ та $z$ -- довільні вектори розмірності $n$.

Якщо ці вектори обираються у вигляді

$$y = \Delta x^{(k)}, \quad z = \mathbf{h}(x^{(k)})\Delta g^{(k)}, \quad (208)$$

матриця $\mathbf{h}(x^{(k+1)})$ обчислюється за допомогою алгоритму Девідона--Флетчера--Пауела. Згідно з цим методом матриця $\Delta \mathbf{h}(x)$ має ранг, що дорівнює 2. $\mathbf{h}(x)$ обчислюється таким чином, щоб для квадратичної функції мети після $n$ кроків вона дорівнювала $H^{-1}$. $\mathbf{h}(x^{(0)}) = E$, але вона може бути й довільною симетричною позитивно визначеною матрицею, так що початковий напрямок мінімізації є напрямок найскорішого спуску.

Оцінка елементів матриці $H^{-1}$ у точці $x^*$ екстремуму буде тим кращою, чим кращою порівнянно з одиничною матрицею ми оберемо початкову матрицю $\mathbf{h}(x^{(0)})$. Для цього можна обирати початкову матрицю шляхом обчислення у початковій точці значень часткових похідних або їхніх скіченно-різницевих наближень. Але перевага віддається першому засобові вибору початкової матриці.

У ході оптимізації має місце поступовий перехід від градієнтного напрямку до ньютоновського. При цьому використовуються переваги кожного з цих двох методів на

відповідному етапі. Враховуючи ((79)), отримуємо наступний алгоритм.

$$\begin{aligned}\mathbf{h}^{(k+1)} &= \mathbf{h}^{(k)} + A^{(k)} - B^{(k)} = \\ &= \mathbf{h}^{(k)} + \frac{(\Delta x^{(k)})(\Delta x^{(k)})^T}{(\Delta x^{(k)})^T(\Delta g^{(k)})} - \frac{\mathbf{h}^{(k)}(\Delta g^{(k)})(\Delta g^{(k)})^T(\mathbf{h}^{(k)})^T}{(\Delta g^{(k)})^T\mathbf{h}^{(k)}(\Delta g^{(k)})}.\end{aligned} \tag{209}$$

Рекурентне співвідношення (74) буде задовільним, якщо:

- похибка при обчисленні $\Delta f(x^{(k)})$ невелика;
- $\mathbf{h}(x^{(k)})$ не стає "поганою".

Роль матриці $A^{(k)}$ у формулі (74) полягає у забезпеченні того, щоб $\mathbf{h} \to H^{-1}$, матриця $B^{(k)}$ забезпечує додатню визначеність $\mathbf{h}^{(k+1)})$.

Запишемо формулу (74) для кількох етапів, починаючи з $\mathbf{h}(x^{(0)})$:

$$\mathbf{h}^{(1)} = E + A^{(0)} - B^{(0)},$$
$$\mathbf{h}^{(2)} = \mathbf{h}^{(1)} + A^{(1)} - B^{(1)} = E + A^{(0)} + A^{(1)} - (B^{(0)} + B^{(1)});$$
$$\dots$$
$$\mathbf{h}^{(k+1)} = E + \sum_{i=1}^{k} A^{(i)} - \sum_{i=1}^{k} B^{(i)}.$$

У разі квадратичної функції мети сума матриць $A^{(i)}$ має дорівнювати $H^{-1}$ при $k = n-1$, а сума матриць $B^{(i)}$ будується таким чином, щоб вона скоротилась із матрицею, що обирається як початкова матриця $\mathbf{h}^{(0)}$, яка у нашому випадку дорівнює $E$. Таким чином, у разі квадратичної форми метод Девідона--Флетчера--Пауела у деякому сенсі у поточному значенні $\mathbf{h}^{(k)}$ відображає усю інформацію, що передує даному значенню.

Відзначимо, що у разі квадратичної функції мети у алгоритмі ДФП використовуються спряжені напрямки, у зв'язку з чим має виконуватись рівність

$$\sum_{i=0}^{n-1} A^{(i)} = \sum_{i=0}^{n-1} \frac{s^{(i)}(s^{(i)})^T}{(s^{(i)})^T H s^{(i)}} = H^{-1}. \tag{210}$$

Чисельник та знаменник дорівнюють відповідно

$$\begin{aligned}(\Delta x^{(k)})(\Delta x^{(k)})^T &= (\lambda^{*(k)} s^{(k)})(\lambda^{*(k)} s^{(k)})^T; \\ (\Delta x^{(k)})^T(\Delta g^{(k)}) &= (\lambda^{*(k)} s^{(k)})^T(H\lambda^{*(k)} s^{(k)}).\end{aligned} \tag{211}$$

Початкова довжина кроку $\lambda^*$ обирається згідно з формулою

$$\lambda^* = \min\left\{1, \frac{2[f(x^{(k)}) - f_0]}{\nabla^T f(x^{(k)})\mathbf{h}^{(k)}\nabla f(x^{(k)})}\right\}, \tag{212}$$

де $f_0$ -- найменше очікуване значення $f(x)$, або ж $\lambda^* = 1$.

Мінімізація функції мети закінчується у точці, в якій при обчисленні вектора $-\mathbf{h}^{(k)}\nabla f(x^{(k)})$ та вектора $-\lambda^{*(k)}\mathbf{h}^{(k)}\nabla f(x^{(k)})$ виконується одна з двох умов:

1. Кожна із складових цих векторів є меншою за деяку заздалегідь задану величину.

2. Довжина (норма), що обчислена для кожного з цих векторів у точці мінінмуму, менше за деяку заздалегідь задану величину.

Згідно з вищевикладеним маємо наступне рекурентне співвідношення для визначення наступного наближення до точки мінімуму функції мети.

$$x^{(k+1)} = x^{(k)} - \lambda^{*(k)}\mathbf{h}^{(k)}\nabla f(x^{(k)}) = x^{(k)} - \lambda^{*(k)}\mathbf{h}^{(k)}\left[\frac{\partial f(x^{(k)})}{\partial x_i^{(k)}}\right]. \qquad (213)$$

Матриця $\mathbf{h}^{(k)}$ визначається за рівнянням (74).

## 3.7.1. Реалізація методу ДФП

Для реалізації методу ДФП необхідно:

1. Обрати або обчислити за яким-небудь методом початкову точку $x^{(0)}$.
2. Обчислити у цій точці

$$\nabla f(x^{(0)}) = \begin{bmatrix} df_1 = \dfrac{\partial f(x^{(0)})}{\partial x_1^{(0)}} \\ \cdots\cdots\cdots \\ df_n = \dfrac{\partial f(x^{(0)})}{\partial x_n^{(0)}} \end{bmatrix}.$$

3. Обрати $\mathbf{h}^{(0)} = E$.
4. Визначити

$$\begin{bmatrix} x_1^{(1)} \\ \cdots \\ x_n^{(1)} \end{bmatrix} = \begin{bmatrix} x_1^{(0)} \\ \cdots \\ x_n^{(0)} \end{bmatrix} - \lambda^{*(0)} \begin{bmatrix} 1 & 0 & \cdots & 0 \\ . & . & \cdots & . \\ 0 & 0 & \cdots & 1 \end{bmatrix} \begin{bmatrix} df_1 \\ \cdots \\ df_n \end{bmatrix}.$$

5. Обчислити

$$\Delta g^{(k)} = \begin{bmatrix} \dfrac{\partial f(x^{(1)})}{\partial x_1^{(1)}} - \dfrac{\partial f(x^{(0)})}{\partial x_1^{(0)}} \\ \cdots\cdots\cdots \\ \dfrac{\partial f(x^{(1)})}{\partial x_1^{(1)}} - \dfrac{\partial f(x^{(0)})}{\partial x_n^{(0)}} \end{bmatrix}.$$

6. Визначити

$$A^{(k)} = \sum_{i=0}^{k} \frac{(\Delta x^{(i)})(\Delta x^{(i)})^T}{(\Delta x^{(i)})^T(\Delta g^{(i)})}.$$

7.

$$B^{(k)} = \sum_{i=0}^{k} \frac{\mathbf{h}^{(i)}(\Delta g^{(i)})(\Delta g^{(i)})^T(\mathbf{h}^{(i)})^T}{(\Delta g^{(i)})^T\mathbf{h}^{(i)}(\Delta g^{(i)})}.$$

8. Визначити

$$\mathbf{h}^{(k+1)} = \mathbf{h}^{(k)} + A^{(k)} - B^{(k)}.$$

9. $x^{(k+1)} = x^{(k)} - \lambda^{*(k)}\mathbf{h}^{(k)}\nabla f(x^{(k)}) = x^{(k)} - \lambda^{*(k)}\mathbf{h}^{(k)}\left[\dfrac{\partial f(x^{(k)})}{\partial x_i^{(k)}}\right].$

10. Якщо $\| \nabla f(x^{(k+1)} \| < \varepsilon$, то процес мінімізації завершено, якшо ні -- то повторити п.п.5--9.

11. Після виконання $n$ ітерацій у тому випадку, якщо потрібну точність ще не досягнуто, виконується "оновлення" методу шляхом привласнення метриці $H = E$.

*Приклад.*

Проілюструємо звстосування методу ДФП на прикладі мінімізації

$$f(x) = 4(x_1 - 5)^2 + (x_2 - 6)^2.$$

Згідно зі схемою реалізації маємо:

1. Нехай $\mathbf{h}^{(0)} = E$.

2.

$$x^{(1)} = \begin{bmatrix} 8 \\ 9 \end{bmatrix}.$$

3.

$$x^{(1)} = x^{(0)} - \lambda^{*(0)} \mathbf{h}^{(0)} \nabla f(x^{(0)})$$

$$= \begin{bmatrix} 8 \\ 9 \end{bmatrix} - \lambda^{*(0)} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 8x_1 - 40 \\ 2x_2 - 12 \end{bmatrix} = \begin{bmatrix} 8 \\ 9 \end{bmatrix} - \lambda^{*(0)} \begin{bmatrix} 24 \\ 0 \end{bmatrix}.$$

4. Знайдемо $\lambda^{*(0)}$. Для цього запишемо

$$f(x^{(1)}) = 4[(8 - 24\lambda^{*(0)}) - 5^2 + [(9 - 6\lambda^{*(0)}) - 6]^2;$$

$$\frac{df(x^{(1)})}{d\lambda^*} = 0 = 51 - 390\lambda^* \rightarrow \lambda^* = 0{,}1307.$$

5.

$$x^{(1)} = \begin{bmatrix} 4{,}862 \\ 8{,}215 \end{bmatrix}; \quad f(x^{(1)}) = 4{,}985.$$

6.

$$\nabla f(x^{(1)}) = \begin{bmatrix} -1{,}108 \\ 4{,}4311 \end{bmatrix}; \nabla g(x^{(0)}) = \begin{bmatrix} -25{,}108 \\ -1{,}569 \end{bmatrix};.$$

7. Обчислимо $\mathbf{h}^{(1)}$:

$$\mathbf{h}^{(1)} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \frac{\begin{bmatrix} -3{,}13 & 0 \\ -0{,}785 & 0 \end{bmatrix} \begin{bmatrix} -3{,}13 & -0{,}785 \\ 0 & 0 \end{bmatrix}}{[-3{,}13 - 0{,}785] \begin{bmatrix} -25{,}108 \\ -1{,}569 \end{bmatrix}}$$

$$- \frac{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} -25{,}108 & 0 \\ -1{,}569 & 0 \end{bmatrix} \begin{bmatrix} -25{,}108 & -1{,}569 \\ 0 & 0 \end{bmatrix}}{[-25{,}108 - 1{,}569] \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} -25{,}108 \\ -1{,}569 \end{bmatrix}} = \begin{bmatrix} 0{,}1270 & -0{,}0315 \\ -0{,}0315 & 1{,}0038 \end{bmatrix}.$$

8. Тепер згідно з п.3 обчислимо

$$\begin{bmatrix} x_1^{(2)} \\ x_2^{(2)} \end{bmatrix} = \begin{bmatrix} 4{,}862 \\ 8{,}215 \end{bmatrix} - \lambda^{*(1)} \begin{bmatrix} 0{,}1270 & -0{,}0315 \\ -0{,}0315 & 1{,}0038 \end{bmatrix} \begin{bmatrix} -1{,}108 \\ 4{,}431 \end{bmatrix} = \begin{bmatrix} 5{,}000 \\ 6{,}000 \end{bmatrix};$$

$$\lambda^{*(1)} = 0,4942.$$

9. Знову обчислюємо

$$\nabla f(x^{(2)}) = \begin{bmatrix} 0,381\cdot 10^{-6} \\ 0,255\cdot 10^{-8} \end{bmatrix};;\ \ f(x^{(2)}) = 0,905\cdot 10^{-4}.$$

Отжу, точку мінімуму знайдено за дві ітерації: $x_1^* = 5$, $x_2^* = 6$.

## 3.8. Контрольні запитання та задачі

1. Чому квадратичні функції використовуються як основа для побудови алгоритмів нелінійної оптимізації?

2. Чому додатна визначеність матриці $A^{(k)}$ є необхідною умовою при розв'язанні задач мінімізації за допомогою квазіньютоновських методів?

3. Чи можливе одержання однакових точок при використанні методів Девідона--Флетчера --Пауела та Флетчера--Рівса для розв'язання задачі із квадратичною функцією мети, якщо в обох випадках початкова точка одна й та ж? Якщо можливе, то чому (за яких умов)? Якщо ні, то чому?

4. Поясніть поняття квадратичної збіжності. Вкажіть один алгоритм, що має властивість квадратичної збіжності, та один алгоритм, який цієї властивості не має.

5. Покажіть, що функція

$$F(x) = 3x_1^2 + 2x_2^2 + x_3^2 - 2x_1x_3 + 2x_2x_3 - 6x_1 - 4x_2 - 2x_3$$

є випукла.

6. Знайдіть та класифікуйте стаціонарні точки функції

$$f(x) = x_1^2 + 4x_1x_2^2 - 10x_1x_2 + x_2^2.$$

7. Використайте метод Гауса-Жордана для перетворення такої квадратичної форми до канонічного вигляду:

$$Q(x) = x_1^2 + 2x_1x_2 + 4x_1x_3 + 3x_3^2 + 2x_2x_3 + 5x_3^2.$$

Покажіть, що ця квадратична форма додатньо визначена.

8. Нехай в точці $x = x\nabla f(x) = 0$. Що можна сказати про точку $x$, якщо

а) $f()$ -- випукла функція;

б) $f()$ -- угнута функція;

в) $\nabla^2 f(x)$ -- невизначена матриця;

г) $\nabla^2 f(x)$ -- додатно визначена матриця;

д) $\nabla^2 f(x)$ -- від'ємно визначена матриця.

9. Сумарні витрати на експлуатацію газопроводу протягом року визначаються функцією

$$C(D, P_1, L, r) = 7,84D^2P_1 + 450000 + 36900D + \frac{10^6}{L}96,57 + 772(r^{0,2} - 1),$$

де $D$ -- внутрішній діаметр труб (см); $L$ -- відстань між компресорними станціями (км); $r = P_1/P_2$ -- відношення тиску на виході та вході компресора.

Припустимо, що питомі витрати газу можна описати функцією

$$Q = 3,4\sqrt{\frac{(P_1^2 - P_2^2)D^5}{fL}} \text{ì}^3/\text{ãîä},$$

де $f = 0,008/\sqrt[3]{D}$ - коефіцієнт тертя. Нехай витрати газ у складають $100 \cdot 10^6 m^3$ /доба. Використайте метод сполучених напрямків Пауела і знайдіть значення параметрів системи, яким відповідає мінімум питомих сумарних експлуатаційних витрат.

10. Задані функція $q(x) = 8x_1^2 + 4x_1x_2 + 5x_2^2$ , початкова точка $x^{(0)} = [10\,10]^T$ та два лінійно незалежні напрямки

$$d^{(0)} = \nabla q(x^{(0)}) = [200\,140]^T,\ d^{(1)} = [7\ -10]^T.$$

Визначіть новий напрям пошуку, сполучений із $d^{(0)}$. Використайте ці напрямки для пошуку $x^*$ за методом сполучених напрямків Пауела, а потім -- за методом Флетчера--Рівса.

11. Знайдіть напрямок, ортогональний вектору

$$s^{(1)} = \left[\frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{3}}\right]^T,\ x^{(0)} = [0, 0, 0]^T.$$

Знайдіть також напрямок $s^{(2)}$, сполучений із $s^{(1)}$ у цій точці за умови, що функція мети

$$f(x) = x_1 + 2x_2^2 - x_1x_2 + 3x_1^2 - 2x_1x_3 + x_3^2.$$

12. Визначіть та класифікуйте стаціонарні точки функції

$$f(x) = x_1^2 - x_1x_2 + x_2^2 - 2x_1 + 3_2 - 4.$$

13. Виконайте три ітерації згідно з методами Ньютона-Рафсона та Девідона-Флетчера-Пауела для мінімізації функції Пауела

$$f(x) = (x_1 + 10x_2)^2 + 5(x_3 - x_1)^2 + (x_2 - 2x_3)^4 + 10(x_1 - x_4)^2.$$

при $x^{(0)} = [3, -1, 0, 1]^T$ .

14. Знайдіть максимум функції мети

$$f(x) = x_1^3 \exp[x_2 - x_1^2 - 10(x_1 - x_2)^2].$$

Порівняйте траєкторії оптимізації у просторі $x$ при використанні методів:

найскорішого спуску;
методу Ньютона;
методу змінної метрики;
методу Флетчера--Рівса.

17. Починаючи з точки $x = [1, -2, 3]^T$ , визначіть точку $x^{(k+1)}$ для функції мети

а) $f(x) = x_1^2 + x_2^2 + x_3^2$;

б) $f(x) = 2x_1^2 + 2x_1x_2 + 3x_2^2 + x_3$;

в) $f(x) = x_1^2 + x_2^2 - x_1 - x_3 + 4$

із застосуванням:

методу сполученого градієнта.
методу Ньютона.

# Розділ 4. Задачі із обмеженнями

Задачі мінімізації функцій з обмеженнями у вигляді рівностей та нерівностей належать (як і задачі без обмежень) до задач нелінійного програмування.

Формально задача нелінійного програмування може бути сформульована наступним чином:

мінімізувати

$$f(x), \quad x \in E^n \tag{214}$$

при $m$ лінійних та (або) нелінійних обмеженнях у вигляді рівностей

$$h_j(x) = 0, \quad j = \overline{1, m}, \tag{215}$$

та $(p - m)$ лінійних та (або) нелінійних обмеженнях у вигляді нерівностей

$$g_j(x) \geq 0, \quad j = \overline{m+1, p}. \tag{216}$$

У деяких випадках обмеження у вигляді рівностей можна явно розв'язати відносно обраних змінних, які потім виключаються із задачі як незалежні змінні й у задачі залишаються тільки обмеження у вигляді нерівностей, але у загальному випадку ці обмеження не можуть бути розв'язані.

Розглянемо алгоритми пошуку екстремуму нелінійної функції при нелінійних обмеженнях. Найпростішим з них, що використовується для розв'язку задач із обмеженнями у вигляді рівностей, є метод множників Лагранжа. У більшості випадків алгоритми оптимізації являють собою різноманітні засоби втілення двох підходів до організації пошуку умовного екстремуму. Перший з них полягає у тому, щоб безпосередньо контролювати дотримання обмеження задачі й рухатись до її оптимуму за послідовністю допустимих або ``майже допустимих" точок із монотонно убуваючими (монотонно зростаючими у разі пошуку максимуму) значеннями функції мети. Відповідні алгоритми мають назву *методів спуску*.

Другий підхід полягає у приведенні задачі на екстремум за наявності обмежень до послідовності задач безумовної оптимізації допоміжних функцій, що конструюються спеціальним чином. Ці функції називаються *штрафними*, а алгоритми, що їх використовують, *методами штрафних функцій*.

## 4.1. Методи спуску

### 4.1.1. Метод проекції градієнту

Цей метод призначений для розв'язку задач виду

$$\begin{aligned} &\min f(x), \\ &\varphi_i(x) \leq 0, \qquad i = \overline{1, m}, \\ &\varphi_i(x) = 0, \qquad i = \overline{m+1, l}, \end{aligned} \tag{217}$$

де $f, \varphi_i$ - неперервно диференційовні функції, й є прямим узагальненням методу найскорішого спуску.

Різниця полягає у тому, що при пошуку безумовного екстремуму напрямок

найскорішого убування функції, що мінімізується, є антиградієнт, а у методі проекції градієнтний напрямок визначається з урахуванням обмежень й отримується внаслідок ортогонального проектування антиградієнту на деяке лінійне різноманіття, що апроксимує ділянку межі допустимої області, ``паралельно" якому буде здійснено крок на черговій ітерації. Оскільки межа є нелінійна, цей крок, взагалі кажучи, виведе за межі допустимої множини, незважаючи на те, що початкова точка належить цій множині. Таким чином, у методі проекції градієнту можливий рух за недопустимими траєкторіями. Однак ступінь порушення обмежень суворо контролюється й зберігається малою за рахунок коригування та обмеження довжин кроків.

Загальна схема однієї ітерації алгоритму полягає у наступному.

• відокремлюються обмеження задачі, що формують межу допустимої множини $\Omega$ в околі поточної точки $x^{(k)}$ й за функціями обмежень будується многогранник $K$, що апроксимує цю множину в околі точки $x^{(k)}$;

• серед векторів, кроки вздовж яких не виводять із $K$, обирається вектор $p^{(k)}$, найближчий до антиградієнту $-f'(x^{(k)})$ (ці вектори утворюють многогранний конус, й вибір $p^{(k)}$ є проектуванням антиградієнту функції мети на цей конус);

• точка $x^{(k)}$ проектується (це й є коригування) на лінійне різноманіття, що утворює грань $K$, у якій лежить вектор $p^{(k)}$;

• функція мети мінімізується на промені, який виходить з проекції точки $x^{(k)}$, що вже знайдена, та має напрямок вздовж вектору $x^{(k)}$ за умови збереження обмежень задачі з урахуванням необхідної точності.

Запровадимо для задачі (4) додатне число $\varepsilon$ та вважатимемо, що точка $x^{(k)}$ задовольняє умовам задачі із достатньою точністю, якщо

$$\begin{aligned} &\varphi_i(x) \le \varepsilon, && i = \overline{1, m}, \\ &|\varphi_i(x)| \le \varepsilon, && i = \overline{m+1, l}. \end{aligned} \tag{218}$$

Нехай у точці $x^{(k)}$ ці нерівності виконуються. Позначимо $I$ набір індексів $i, 1 \le i \le l$ таких, що $|\varphi_i(x)| \le \varepsilon$. У цей набір увійдуть номери усіх обмежень - рівностей, а можливо й номери деяких обмежень-нерівностей. Оскільки усі інші обмеження виконуються в $x^{(k)}$ із запасом, межа допустимої множини задачі (5) в околі $x^{(k)}$ буде визначатись тільки функціями $\varphi_i(x), i \in I$. При цьому апроксимуючий многогранник $K$ задається умовами:

$$\begin{aligned} &(A_i, x - x^{(k)}) + \varphi_i(x^{(k)}) \le 0; \quad i \le m; i \in I \\ &(A_i, x - x^{(k)}) + \varphi_i(x^{(k)}) \le 0, \quad i \ge m+1, \\ &A_i = [\frac{\partial \varphi_i(x^{(k)})}{\partial x_1}, \quad \ldots, \frac{\partial \varphi_i(x^{(k)})}{\partial x_n}]. \end{aligned}$$

Відповідно, вектори $p$, кроки вздовж яких не виводять із множини $K$, мають задовольняти збереження співвідношень

$$\begin{aligned} &(A_i, p) \le 0; && i \le m; i \in I \\ &(A_i, p) = 0, && i \ge m+1, \end{aligned}$$

Необхідно вибрати той, що менше за інші відрізняється від антиградієнта $(-f'(x^{(k)})$, тобто розв'язати задачу квадратичного програмування

$$\begin{aligned} &\min\left[(f'(x^{(k)})+p),(f'(x^{(k)})+p)\right], \\ &(A_i, p) \le 0, \quad i \le m; i \in I, \\ &(A_i, p) = 0, \quad i \ge m+1. \end{aligned} \tag{219}$$

Якщо вектори $A_i, i \in I$ лінійно незалежні, а на практиці це майже завжди виконується, то для розв'язку задачі ((6)) можна використовувати алгоритм, на $s$ -у кроці якого розв'язується задача вигляду

$$\begin{aligned} &\min\{F(p) \quad = \left((f'(x^{(k)})+p),(f'(x^{(k)})+p)\right)\}, \\ &(A_i, p) \quad = 0, \quad i \in I_s \subset I. \end{aligned} \tag{220}$$

Множина $I_s$ при $s \ge 1$ містить усі $i = \overline{m+1, l}$ й визначається на попередньому кроці, а $I_0 = I$ . За правилом множників Лагранжа маємо рівняння:

$$\begin{aligned} &\frac{\partial L(p^*, \lambda^*)}{\partial p} \quad = 2f'(x^{(k)}) + 2p^* + \sum_{i \in I_s} \lambda_i^* A_i = 2f'(x^{(k)}) + 2p^* + A\lambda^* = 0; \\ &\frac{\partial L(p^*, \lambda^*)}{\partial \lambda^*} \quad = A^T p^* = 0. \end{aligned} \tag{221}$$

Помножуючи рівняння (8) зліва на $A^T$ , отримаємо

$$2A^T f'(x^{(k)}) + A^T A\lambda^* = 0.$$

Звідси маємо

$$\begin{aligned} &\lambda^* \quad = -2\left(A^T A\right)^{-1} A^T f'(x^{(k)}; \\ &p^* \quad = \left[A\left(A^T A\right)^{-1} A^T - E\right] f'(x^{(k)}. \end{aligned} \tag{222}$$

Згідно з викладеною схемою методу проекції градієнту перед тим, як здійснити крок вздовж вектора $p_k$ , що отримується як розв'язок задачі (7), треба спроектувати точку на лінійне різноманіття, що задане рівнянням

$$(A_i, x - x_0) + \varphi_i(x_k) = 0, \quad i \in I_k, \tag{223}$$

де $I_k$ -- множина індексів $i \in I$ , для яких $(A_i, p_k) = 0$ .

Для проекції $x^{(k)}$ точки $x_k$ має місце формула

$$x^{(k)} = x^{(k)} + \bar{A}(\bar{A}^T A)^{-1} \bar{\varphi} x^{(k)}). \tag{224}$$

У цій формулі $\bar{A}, \bar{\varphi}(x^{(k)})$ -- матриця та вектор, що складені із стовпців $A_i$ та компонент $\varphi(x^{(k)})$ з індексами із множини $I_k$ . Після визначення точки $x^{(k)}$ для завершення ітерації методу проекції градієнта залишається розв'язати задачу мінімізації

$$\begin{aligned} &\min_{\alpha \ge 0} f(x^{(k)} + \alpha p^{(k)}), \\ &\varphi_i(x^{(k)} + \alpha p^{(k)}) \le \varepsilon, \quad i = \overline{1, m}, \\ &|\varphi_i(x^{(k)} + \alpha p^{(k)})| \le \varepsilon, \quad i = \overline{m+1, l}. \end{aligned} \tag{225}$$

Як наслідок, отримаємо точку $x^{(k+1)} = x^{(k)} + \alpha_k p^{(k)}$, у якій усе повторюється спочатку.

*Приклад.* Розв'язати задачу

$$f(x) = 2x_1^2 + 2x_2^2 - 2x_1x_2 - 4x_1 - 6x_2 \to \min$$

із обмеженнями

$$x_1 + x_2 \le 2,$$
$$x_1 + 5x_2 \le 5,$$
$$x_1 \ge 0,\ \ x_2 \ge 0.$$

Оберемо як початкове наближення точку $x^{(0)} = [0, 0]^T$.

Перша ітерація. *Пошук напрямку.* У точці $x^{(0)} = [0, 0]^T$ маємо:

$$\nabla f(x^{(0)}) = [-4, -6]^T.$$

$$A_1 = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}; \quad A_2 = \begin{bmatrix} 1 & 1 \\ 1 & 5 \end{bmatrix}.$$

$$P = E - A_1^T (A_1 A_1^T)^{-1} A_1 = A_1 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

Звідси $s^{(0)} = -P\nabla f(x^{(0)} = [0, 0]^T$. Якщо взяти до уваги, що обмеження рівності відсутні, тобто $v = 0$, обчислимо

$$w = u = -(A_1 A_1^T)^{-1} A_1 \nabla f(x^{(0)}) = [-4, -6]^T.$$

Оберемо $u_4 = -6$ і видалимо градієнт, що відповідає 4-му обмеженню у $A_1$:

$$A_1 \to \hat{A}_1 = \begin{bmatrix} -1 & 0 \\ 0 & 0 \end{bmatrix}.$$

$$\hat{P} = E - \hat{A}_1 (\hat{A}_1 \hat{A}_1)^{-1} \hat{A}_1 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}.$$

$$s^{(1)} = -\hat{P}\nabla f(x^{(0)}) = -\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} -4 \\ -6 \end{bmatrix} = \begin{bmatrix} 0 \\ 6 \end{bmatrix}.$$

*Лінійний пошук.* Довільну точка $x^{(1)}$, яку отримано рухом із $x^{(0)}$ у напрямку $s^{(0)}$ можна подати у вигляді

$$x^{(1)} = x^{(0)} + \lambda s^{(0)} = [0, 6\lambda]^T,$$

а відовідне до нього значення функції мети дорівнює $f(x^{(1)}) = 72\lambda^2 - 36\lambda$.

$$\lambda_{\max} = \min\{2/6; 5/30\} = 1/6.$$

Отже, $\lambda_0$ є оптимальний розв'язок задачі

мінімізувати $72\lambda^2 - 36\lambda$

за умови $0 \le \lambda \le 1/6$. Оптимальний розв'язок дорівнює $\lambda_0^{(0)} = 1/6$ і

$$x^{(1)} = x^{(0)} + \lambda_1^{(0)} s^{(0)} = [0, 1]^T.$$

Друга ітерація. *Пошук напрямку.* $\nabla x^{(1)} = [-6, -2]^T$. Крім того, активними є тільки друге і третє обмеження (тобто справджується рівність), отже маємо

$$A_1 = \begin{bmatrix} 1 & 5 \\ -1 & 0 \end{bmatrix}; \quad A_2 = \begin{bmatrix} 1 & 1 \\ 0 & -1 \end{bmatrix}; \; P = A_1 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

Обчислюємо

$$u = -(A)_1 A_1^T)^{-1} A_1 \nabla f(x^{(1)}) = [2/5, -28/3]^T.$$

Оскільки $u_3 < 0$, рядок $[-1, 0]$ із $A_1$ викреслюється:

$$\hat{A}_1 = A_1 = \begin{bmatrix} 1 & 5 \\ 0 & 0 \end{bmatrix}.$$

Матриця проектування і відповідний напрямок визначається таким чином:

$$\hat{P} = \begin{bmatrix} \frac{25}{26} & -\frac{5}{26} \\ -\frac{5}{26} & -\frac{1}{26} \end{bmatrix}; \quad s^{(1)} = \left[\frac{70}{13}, -\frac{14}{13}\right]^T.$$

Оскільки для вибору напрямку довжина $s^{(1)}$ не має значення, можна покласти $s^{(1)} = [5, -1]^T$.

*Лінійний пошук.* $x^{(1)} + \lambda s^{(1)} = [5\lambda; 1-\lambda]^T$, в якій $f(x^{(1)} + \lambda s^{(1)}) = 62\lambda^2 - 28\lambda - 4$. Максимальне значення $\lambda$, для якого точка є допустимою, дорівнює $\lambda_{\max} = \min\{1/4; 1/1\} = 1/4$. Отже, $\lambda_2$ визначається із розв'язання задачі:

мінімізувати $62\lambda^2 - 28\lambda - 4$ за умови $0 \le \lambda \le 1/4$.

Оптимальним є значення $\lambda_2 = 7/31$, і

$$x^{(2)} = \left[\frac{35}{31}, \frac{24}{31}\right]^T.$$

Третя ітерація. *Пошук напрямку*

$$\nabla f(x^{(2)}) = \left[-\frac{32}{31}, -\frac{160}{31}\right]^T.$$

У цій точці активним є друге обмеження, тобто

$$A_1 = \begin{bmatrix} 1 & 5 \\ 0 & 0 \end{bmatrix}.$$

Тоді

$$P = \frac{1}{28}\begin{bmatrix} 25 & -5 \\ -5 & 1 \end{bmatrix}; \quad s^{(2)} = -P\nabla f(x^{(2)}) = [0, 0]^T.$$

$$u = -(A_1 A_1^T)^{-1} A_1 \nabla f(x^{(2)}) = \frac{32}{31} > 0.$$

Отже, точка $x^{(2)}$ -- оптимальна.

## 4.1.2. Метод допустимих напрямків

Алгоритм допустимих напрямків розроблено Г. Заутендейком й призначений для пошуку екстремуму при наявності обмежень у вигляді нерівностей. На відміну від методу

проекції градієнта цей алгоритм розглядає суворо допустимі точки задачі.

Суть методу можливих напрямків полягає у наступному: пошук починається у допустимій точці простору розв'язків й реалізується (при лінеаризованих обмеженнях) за траєкторією, що забезпечує покращання значень функції, але ніколи не виходить за межі допустимої області. У цьому сенсі метод проекції градієнту є однією з модифікацій метода допустимих напрямків. Варто зауважити, що якщо у методі проекції градієнту проекція градієнту $\nabla f(x^{(k)})$ на підмножину обмежень, що містять $x^{(k)}$, задає напрямок пошуку, що відповідає напрямку найскорішого спуску до функції мети, то у методі Заутендейка використовується метрика, що визначається співвідношенням

$$\| \Delta x \| = \max\{|\Delta x_1|, \ldots, |\Delta x_n|\}.$$

Ця метрика визначає напрямок руху, що забезпечує найбільшу мінімізуючу поправку до значення функції мети, не порушуючи будь-якого із обмежень.

Метод Заутендейка дозволяє оперувати як з лінійними, так й з нелінійними обмеженнями. Однак він не дозволяє оперувати з обмеженнями у вигляді рівностей. Лінеарізація функцій виконується шляхом розкладу їх у ряд Тейлора в околі точки $x^{(k)}$. Це призводить до наступної задачі:

мінімізувати $f(x^{(k)}) + \nabla^T f(x^{(k)})(x - x^{(k)}),\ x \in E^n$,

з обмеженнями

$$g_i(x^{(k)}) + \nabla^T g_i(x^{(k)})(x - x^{(k)}) \geq 0,\ i = \overline{1, p}. \qquad (226)$$

Процедура Заутендейка встановлює найбільш сприятливі з можливих напрямків оптимізаційного пошуку (з точки зору отримання найбільшої мінімізуючої поправки до значення $f(x)$ при переміщенні з точки $x^{(k)}$ у точку $x^{(k+1)}$ без порушення якого-небудь з обмежень). Це вирішується шляхом розв'язку задачі лінійного програмування.

Підстановка у задачу замість вектора $x$ його поточного значення $x^{(k+1)}$ призводить до задачі мінімізації

$$f(x^{(k)}) + \lambda^{(k)} \nabla^T f(x^{(k)}) s^{(k)} \qquad (227)$$

з обмеженнями

$$g_i(x^{(k)}) + \lambda^{(k)} \nabla^T g_i(x^{(k)})(s^{(k)}) \geq 0,\ i = \overline{1, p}. \qquad (228)$$

Оскільки $f(x^{(k)})$ та $g_i(x^{(k)}) \geq 0$ є сталими, необхідні та достатні умови реалізації переміщення з точки $x^{(k)}$) у точку $x^{(k+1)}$) у зв'язку з розв'язком задачі (14), (15) мають вигляд:

$$\begin{aligned} \nabla^T f(x^{(k)}) s^{(k)} &< 0, \\ \nabla^T g_i(x^{(k)}) s^{(k)} &\geq 0, \quad i = \overline{1, p}. \end{aligned} \qquad (229)(230)$$

Метод Заутендейка відокремлює допустимий напрямок, що забезпечує отримання найбільшої мінімізуючої поправки до значення функції мети $f(x)$ на кроці від точки $x^{(k)}$ до точки $x^{(k+1)}$ шляхом розв'язку наступної задачі лінійного програмування:

мінімізувати $\nabla^T f(x^{(k)}) s^{(k)}, \quad s \in E^n$, з обмеженнями (17) .

Кожний з етапів реалізації методу Заутендейка має наступну алгоритмічну структуру:

• Нехай $x^{(k)}$ -- деяка допустима точка задачі:

мінімізувати $f(x),\ x \in E^n$,

з обмеженнями

$$g_i(x) \geq 0, \quad i = \overline{1, p}.$$

• Обчислюються градієнти функції $f(x)$ та функцій $g_i(x)$ у точці $x^{(k)}$ й розв'язується задача лінійного програмування будь-яким з відомих алгоритмів лінійного програмування, внаслідок чого відшукується допустимий напрямок $s^{(k)}$.

• Якщо $\nabla^T f(x^{(k)})s^{(k)} < 0$, то визначається максимальна довжина кроку $\lambda^*$, що здійснюється у напрямку $s^{(k)}$ без виходу за межі допустимої області задачі ((34)). При цьому $\lambda^* = \max\{\lambda \mid x^{(k)} + \lambda s^{(k)} \in R\}$. Далі визначається таке значення $\lambda^{(k)}$, що задовольняє умові $0 \leq \lambda^{(k)} \leq \lambda^*$, для якого $f(x^{(k)} + \lambda^{(k)} s^{(k)})$ є мінімальним значенням функції мети у напрямку $s^{(k)}$. Нова точка визначається з $x^{(k+1)} = x^{(k)} + \lambda^{(k)} s^{(k)}$.

• Якщо $\nabla^T f(x^{(k)})s^{(k)} = 0$, оптимізаційний пошук закінчується.

До переваг методу можливих напрямків слід віднести його надійність та універсальність стосовно задач із обмеженнями у вигляді нерівностей. До недоліків слід віднести невисоку швидкість збіжності за кількістю ітерацій.

## 4.2. Методи штрафних функцій

Характерною рисою цієї групи методів є отримання розв'язку задачі з обмеженнями як границі послідовностей розв'язку допоміжних задач безумовної оптимізації допоміжних функцій, що підбираються відповідним чином.

Переваги переходу від задачі мінімізації за наявності обмежень до задачі мінімізації без обмежень, тобто задачі на безумовний екстремум полягають у тому, що в останньому випадку мінімізація може бути здійснена за допомогою значно простіших алгоритміів порівняно із першим випадком.

При застосуванні методів штрафних функцій отримується аксимальний оптимізуючий ефект внаслідок постійного компромісу між необхідністю задовольняти обмеження та процесом мінімізації функції мети. Цей компроміс досягається шляхом привласнення відповідних вагових коефіцієнтів функції мети та функціям-обмеженням.

Методи штрафних функцій поділяються на дві групи: параметричні методи та непараметричні методи. Параметричні методи характеризуються наявністю одного або кількох відповідним чином підібраних параметрів, що входять у штрафну функцію (яка будується за допомогою функцій - обмежень) у якості вагових коефіцієнтів. Параметричні методи розділяються на три категорії: методи внутрішньої точки, методи зовнішньої точки та комбіновані методи. При застосуванні методів внутрішньої точки рівень функції мети утримується на відстані від межі допустимої області за допомогою штрафної функції.

Методи зовнішньої точки навпаки, генерують послідовність точок, що виходять за межі допустимої області, але дають граничний допустимий розв'язок. При цьому штрафна функція не дозволяє вектору $x$ дуже далеко віддалятися від межі допустимої області. У комбінованих методах (використання яких особливо необхідне у випадках, коли обмеження мають вигляд рівностей) у ході мінімізації одні з обмежуючих умов задовольняються, а інші ні, але ж усі умови у межах допуску, що є заданим, виявляються такими, що задовольняються при

досягненні шуканого розв'язку.

Отже, в основу методів штрафних функцій покладено ідею перетворення загальної нелінійної задачі:

мінімізувати

$$f(x),\ x \in E^n \tag{231}$$

при $m$ лінійних та (або) нелінійних обмеженнях у вигляді рівностей

$$h_i(x) = 0,\ i = \overline{1, m} \tag{232}$$

й $(p - m)$ лінійних та (або) нелінійних обмеженнях у вигляді нерівностей

$$g_j(x) = 0,\ j = \overline{m+1, p} \tag{233}$$

у послідовність задач без обмежень шляхом додавання до функції мети однієї або декількох функцій, що задають обмеження, таким чином, щоб обмеження у задачі оптимізації не фігурували. Формально перетворення задачі з обмеженнями у задачу без обмежень здійснюється шляхом переходу від задачі ((36)) --((38)) до задачі мінімізації

$$P(x^{(k)}, r^{(k)}) = f(x^{(k)}) + \sum_{i=1}^{m} r_i^{(k)} H[h_i(x^{(k)})] + \sum_{i=m+1}^{p} r_i^{(k)} G[g_i(x^{(k)})].) \tag{234}$$

$P(x^{(k)}, r^{(k)})$ має назву узагальненої приєднаної функції або *штрафної функції*. У формулі (21) $r_i^{(k)}$ -- вагові коефіцієнти. Функціонали $H[h_i(x^{(k)})]$ та $G(g_i[x^{(k)})]$ обираються з урахуванням конкретних вимог.

Типовими вимогами при виборі функціоналу $G[g_i(x^{(k)})]$ є наступні:

$G[g_i(x^{(k)})] > 0$ при $g_i(x) < 0$ та $G[g_i(x^{(k)})] = 0$ при $g_i(x) \geq 0$. При такому виборі функціоналу $G$ не турбуються, щоб обмежувальні умови виконувались на проміжних етапах обчислювального процесу, а вимагають тільки, щоб ці умови задовольнялись у точці, що визначає шуканий розв'язок.

У випадку обмежень у вигляді рівностей вимагають, щоб $H[h_i(x^{(k)})] \to 0$ при $h_i(x) \to 0$. При цьому зазвичай вважають, що $H[h_i(x)] = h_i^2(x)$.

Можливі й інші засоби вибору функціоналів $G[g_i(x^{(k)})], H[h_i(x^{(k)})]$. При будь-якому виборі вимагають, щоб

$$\begin{aligned} &\lim_{k\to\infty} \sum_{i=m+1}^{p} r_i^{(k)} G[g_i(x^{(k)})] = 0; \\ &\lim_{k\to\infty} \sum_{i=1}^{m} r_i^{(k)} H[h_i(x^{(k)})] = 0; \\ &\lim_{k\to\infty} |P(x^{(k)}, r^{(k)}) - f(x^{(k)})| = 0. \end{aligned} \tag{235}$$

Це означає, що вплив функцій-обмежень, що входять у $P(x^{(k)}, r^{(k)})$, на значення приєднаної функції у ході обчислювального процесу зменшується, й для граничних значень цієї функції зовсім зникає, так що послідовність проміжних значень $P(x^{(k)}, r^{(k)})$ збігається до того ж значення, що й відповідна послідовність значень $f(x^{(k)})$, отже екстремум функції $P(x)$ збігається з екстремумом функції $f(x)$.

### 4.2.1. Метод узагальнених множників Лагранжа

Метод узагальнених множників Лагранжа належить до категорії параметричних методів штрафних функцій, оскільки у ньому функції-обмеження вводяться у структуру модифікованої функції мети разом із деякими змінними параметрами. Для узагальнення методу Лагранжа обмеження у вигляді нерівностей перетворюються у обмеження у вигляді рівностей шляхом запровадження відповідних послаблюючих змінних.

Задача нелінійного програмування і (18)- (20) при цьому набуває наступного вигляду.

мінімізувати $f(x),\ x \in E^n$

з обмеженнями

$$\begin{aligned} & h_i(x) = 0.\ i = \overline{1, m}, \\ & g_i(x) - v_i^2 = 0,\ i = \overline{m+1, p} \end{aligned} \tag{236}$$

Функція Лагранжа набуває вигляду

$$L(x, \mathbf{l}) = f(x) + \sum_{i=m+1}^{p} \lambda_i h_i(x) + \sum_{i=1}^{m} \lambda_i [g_i(x) - v_i^2], \tag{237}$$

де $\lambda_i$ -- невід'ємні вагові коефіцієнти, які можна ототожнити із множниками Лагранжа.

Для того, щоб $x^*$ було розв'язком загальної задачі нелінійного програмування (18)--(20), необхідно та достатньо, щоб:

1) $f(x^*)$ була випуклою,

2) в околі $x^*$ були випуклі обмеження задачі та

3) у точці $x^*$ задовольнялась наступна система рівнянь, що визначає стаціонарний розв'язок (23):

$$\begin{aligned} & \frac{\partial L(x^*)}{\partial x_i} = 0 \quad \text{ïðèi} = \overline{1, n}; \\ & \frac{\partial L(x^*)}{\partial \lambda_i} = 0 \quad \text{ïðèi} = \overline{1, m}; \\ & \frac{\partial L(x^*)}{\partial \lambda_i} = 2\lambda_i v_i = 0 \quad \text{ïðèi} = \overline{m+1, p}; \\ & \lambda_i \geq 0. \end{aligned} \tag{238}$$

Таким чином, умовний екстремум функції $f(x)$ має місце у стаціонарній точці для $L(x, \mathbf{l}, v)$, зокрема, у сідловій точці $(x, \mathbf{l}, v)$ простору, таким чином, що задача із обмеженнями перетрворюється у задачу визначення сідлової точки за відсутності обмежень.

Алгоритмична схема цього методу полягає у формуванні функції Лагранжа (24), запису системи рівнянь (24), що визначає стаціонарну точку цієї функції та пошуку розв'язку цієї системи рівнянь за допомогою будь-якого з методів розв'язку систем алгебраїчних рівнянь.

*Приклад.*

Мінімізуємо функцію $y = x_1 x_2,\ x \in E^2$

із обмеженнями

$$g_1(x): 9 - x_1^2 - x_2^2 \geq 0.$$

Узагальнена функція Лагранжа має вигляд:

$$L(x, \lambda) = x_1 x_2 - \lambda_1 (9 - x_1^2 - x_2^2 - v_1^2).$$

Необхідні умови існування екстремуму:

$$\frac{\partial L}{\partial x_1} = x_2 + 2\lambda_1 x_1 = 0;$$

$$\frac{\partial L}{\partial x_2} = x_1 + 2\lambda_1 x_2 = 0;$$

$$\frac{\partial L}{\partial \lambda_1} = 9 - x_1^2 - x_2^2 - v_1^1 = 0;$$

$$\frac{\partial L}{\partial v_1} = 2\lambda_1 v_1 = 0.$$

Ця система рівнянь містить 5 невідомих. Нехай вільним буде $\lambda_1$. Якщо відняти друге рівняння із першого, отримаємо

$$2\lambda_1 (x_1 - x_2) = 0 \rightarrow \lambda_1 = 0, \; x_1 = x_2.$$

Якщо $\lambda_1 = 0$ то $x_1 = x_2 = 0$, тоді $v_1 = 3$. Отримуємо сідлову точку (розв'язок -- центр кола). Якщо ж $\lambda_1 \neq 0$, при значенні $\lambda_1 = 0,5$ маємо $v_1 = 0$, $x_1 = \pm 2,121$; $x_2 = \mp 2,121$. Значення функції мети дорівнює -4,5, тобто, функція має мінімум. Якщо $\lambda_1 = -0,5$, маємо $v_1 = 0$, $x_1 = \pm 2,121$; $x_2 = \pm 2,121$. Значення функції мети дорівнює 4,5, тобто, функція має максимум.

До недоліків викладеного методу слід віднести ту обставину, що у загальному випадку маємо розв'язувати систему нелінійних алгебраїчних рівнянь. При цьому, враховучи, що проблеми, які виникають при розв'язанні цієї задачі часто є складнішими за задачу оптимізації, цей метод слід застосовувти для задач із невеликою кількістю змінних.

## 4.2.2. Метод послідовної безумовної мінімізації (МПБМ)

Метод послідовної безумовної мінімізації (МПБМ) є комбінованим методом штрафних функцій, коли прямування до точки мінімуму здійснюється з одного боку за методом внутрішніх точок, а з іншого -- за методом зовнішніх точок. Метод полягає у тому, що шукається розв'язок деякої послідовності задач без обмежень, при цьому у граничному випадку отримується мінімум початкової задачі нелінійного програмування. Початкова задача нелінійного програмування перетворюється у послідовність задач без обмежень шляхом побудови $P$-функції вигляду:

$$P(x^{(k)}, r^{(k)}) = f(x^{(k)}) + \frac{1}{\sqrt{r^{(k)}}} \sum_{i=1}^{m} h_i^2(x^{(k)}) + r^{(k)} \sum_{j=m+1}^{p} \frac{1}{g_i(x^{(k)})}, \qquad (239)$$

де значення вагових коефіцієнтів $r$ додатні й утворюють монотонно спадаючу послідовність $\{r \mid r^{(0)} > r^{(1)} > \ldots > 0\}$.

Функціонал $G$ у правій частині останнього рівняння можна обирати, зокрема, у вигляді

$$G[g(x^{(k)})] = \sum_{i=1}^{p} \min\{0, g_i(x^{(k)})\}^2,$$

або у вигляді

$$G[g(x^{(k)})] = -\sum_{i=1}^{p} \ln g_i(x^{(k)}).$$

Тоді замість (26) отримаємо штрафну функцію

$$P(x^{(k)}, r^{(k)}) = f(x^{(k)}) + \frac{1}{r^{(k)}} \sum_{i=1}^{m} h_i^2(x^{(k)}) - r^{(k)} \sum_{j=m+1}^{p} \ln g_i(x^{(k)}), \qquad (240)$$

Процедура мінімізації функцій (26) або (27) починається з внутрішньої (або межової) точки, тобто, з точки $x$, в якій усі обмежувальні умови у вигляді нерівностей виконуються. Після обчислення $r^{(0)}$ точка $x^{(1)}$ обчислюється шляхом мінімізації $P(x, r^{(0)})$. Потім обчислюється $r^{(1)}$ тощо.

При цьому виникає кілька проблем. По перше, асоційована із $P$-функцією матриця Гессе при наближенні до екстремуму починає вести себе все гірше; у зв'язку з цією обставиною напрямки оптимізаційного пошуку можуть виявитися помилковими. По друге, швидкість збіжності залежить від початкового вибору $r^{(0)}$, а також від засобу редуціювання цього параметра.

Можливі три засоби вибору початкового значення $r^{(0)}$:

- $r^{(0)} = 1$.
- Значення $r^{(0)}$ обирається так, щоб воно мінімізувало норму градієнту $P$-функції, тобто

$$r^{(0)} = \frac{-\nabla^T f(x^{(0)}) \nabla R(x^{(0)})}{\| \nabla R(x^{(0)}) \|^2}, \qquad (241)$$

де $x^{(0)}$ -- внутрішня допустима точка, $R(x^{(0)}) = \sum_{i=m+1}^{p} 1/g_i(x^{(0)})$. При такому виборі $r^{(0)}$ враховуються тільки обмеження у вигляді нерівностей.

- Значення $r^{(0)}$ обирається так, щоб воно мінімізувало перевищення функції $P(x, r^{(0)})$ відносно його мінімального значення (цю величину називають виваженим за метрикою градієнтом $P$-функції).

$$r^{(0)} = \sqrt{\frac{\nabla^T f(x^{(0)}) [\nabla^2 R(x^{(0)})]^{-1} \nabla f(x^{(0)})}{\nabla^T R(x^{(0)}) [\nabla^2 R(x^{(0)})]^{-1} \nabla R(x^{(0)})}}, \qquad (242)$$

де $\nabla^2 R(x^{(0)})$ -- матриця Гессе для $R(x^{(0)}) = \sum_{i=m+1}^{p} 1/g_i(x^{(0)})$, що обчислюється у точці $x^{(0)}$. Цим співвідношенням $r^{(0)}$ задається точно, якщо $P$-функція є квадратична.

Співвідношення (28) може бути застосовано, якщо $\nabla^2 R(x^{(0)}) \neq 0$.

З наведених засобів вибору $r^{(0)}$ із практичних міркувань найзручнішим є перший. Але для того, щоб досягти найбільшої точності, слід віддавати перевагу співвідношенню (28), незважаючи на те, що за досить великого значення $r^{(0)}$ зростуть витрати часу за рахунок

збільшення кількості ітераций. Співвідношення (27) використовується у тому випадку, коли відсутні дані про другі похідні функції $R$. Якщо внаслідок обчислень за формулами (27) або (28) виявиться, що $r^{(0)} \le 0$, оптимізаційний пошук виконується за одним з методів, що викладено у попередньому розділі (наприклад, за методом Ньютона). При цьому $r^{(0)}$ обчислюється у кожній новій точці $x^{(k)}$, доки не буде отримане додатне значення $r^{(0)}$. Далі оптимізаційний пошук продовжується за методом МПБМ. Якщо ж значення $r^{(0)} \le 0$ при досягненні мінімуму функції мети, це означає, що точка $x^*$ -- внутрішня точка, й пошук закінчується.

**Теорема (про збіжність МПБМ).**

*Якщо задача нелінійного програмування задовольняє умовам:*

• задача має допустиму область;

• функції $f(x)$, $h_1(x),\ldots,h_m(x)$, $g_{m+1}(x),\ldots,g_p(x)$ мають бути подвійно неперервно диференційовними (при використанні методу, що ґрунтується на використанні других похідних);

• для будь-якого скінченного $q$ та $\forall r^{(0)} \ge 0$ множина точок $x \in S^*$, що задовольняє нерівності

$$f(x) + \frac{1}{\sqrt{r^{(k)}}} \sum_{i=1}^{m} h_i^2(x) \le q$$

має бути обмеженою ($S^*$ -- містить усі точки $x$ такі, що $g_i(x) \ge 0$);

• функція мети $f(x)$ має бути випуклою, а функції $h_i(x)$ $(i = \overline{1,m})$ мають бути лінійними (точніше, має бути випуклою функція $\sum_{i=1}^{m} h_i^2(x)$);

• функції $g_j(x)$ ($j = \overline{m+1, p}$) мають бути увігнутими;

• матриця Гессе для $P$-функції відносно $x$ має не дорівнювати нулю для усіх точок $x \in S^*$,

то:

а) кожна функція $P(x, r^{(k)})$ має мінімум у деякій точці $x(r^{(k)}) \in S^*$ й

б) $\lim\limits_{r^{(k)} \to \infty} P(x, r^{(k)}) = \min f(x) = f(x^*)$. Крім того, функція $P$ є випуклою.

Якщо умова 6) не виконується, вибір $r^{(0)}$ здійснюється за формулою (27) або $r^{(0)} = 1$.

Якщо не виконується умова 4, тобто функції $h_i(x)$ є нелінійними, треба розв'язувати двоїсту задачу нелінійного програмування. Задача (18)--(20) переформулюється наступним чином:

мінімізувати $f(x), \quad x \in E^n$

з обмеженнями

$$\begin{aligned} h_i^2(x) &\ge 0, \quad i = \overline{1,m}, \\ g_j(x) &\ge 0, \quad j = \overline{m+1, p}. \end{aligned} \tag{243}$$

Двоїстою до цієї задачі буде наступна:

мінімзувати

$$E(x,u,w) = f(x) - \sum_{j=m+1}^{p} u_j g_j(x) + \sum_{i=1}^{m} w_j h_j^2(x) \tag{244}$$

з обмеженнями

$$\begin{aligned} &\nabla_x E(x,u,w) \quad = 0, \\ &u_i \geq 0, \qquad i = \overline{m+1, p}; \\ &w_i \geq 0, \qquad i = \overline{1, m}. \end{aligned} \tag{245}$$

**Реализація МПБМ**

У загальному випадку необхідно виконати такі дії.

• Обрати початкову точку $x^{(0)} \in S^*$ (тобто таку точку, для якої $g_i(x^{(0)}) \geq 0$, $i = \overline{m+1, p}$. Інакше кажучи, початкова точка $x^{(0)}$ має бути внутрішньою точкою допустимої області. Якщо при перевірці цієї умови виявиться, що для деяких з обмежень ця точка є зовнішньою, то для цих обмежень формується допоміжна $P'$-функція для кожного з цих обмежень й здійснюється пошук її мінімуму.

$$P'(x) = r \sum_{k=i+1}^{n} \frac{1}{g_k(x)} - g_i(x), \tag{246}$$

де $g_i(x)$ -- обмеження, для якого не виконується умова $g_i(x) \geq 0$.

$$\begin{aligned} \nabla P'(x) \quad &= -\nabla g_i(x) - r \sum_{k=i+1}^{n} \frac{\nabla g_k(x)}{g_k^2(x)}; \\ \nabla^2 P'(x) \quad &= -\nabla^2 g_i(x) - r \sum_{k=i+1}^{n} \frac{\nabla^2 g_k(x)}{g_k^2(x)} + 2r \sum_{k=i+1}^{n} \frac{(\nabla g_k(x))^2}{g_k^3(x)}. \end{aligned} \tag{247)(248}$$

Визначити початкове значення $r^{(0)}$ згідно з формулою (27) або (28). Залежно від обраної формули пошук $r^{(0)}$ здійснюється шляхом мінімізації $P$-функції за методом Ньютона або ДФП чи іншим методом першого порядку.

З метою реалізації алгоритму мінімізації $P$-функції запишемо градієнти цієї функції:

$$\nabla P(x) = \nabla f(x) + \frac{2}{\sqrt{r}} \sum_{k=1}^{m} h_k(x) \nabla h_k(x) - r \sum_{k=m+1}^{p} \frac{\nabla g_k(x)}{g_k^2(x)}; \tag{249}$$

$$\begin{aligned} \nabla^2 P(x) = \nabla^2 f(x) + \frac{2}{\sqrt{r}} \sum_{k=1}^{m} \Big[ h_k(x) \nabla^2 h_k(x) + (\nabla h_k(x))^2 \Big] \\ - r \sum_{k=m+1}^{p} \left[ \frac{\nabla^2 g_k(x)}{g_k^2(x)} - 2 \sum_{k=i+1}^{n} \frac{(\nabla g_k(x))^2}{g_k^3(x)} \right]. \end{aligned} \tag{250}$$

• Для визначеного початкового $r^{(0)}$ здійснюється пошук мінімального значення $P$-функції (25). Завершення пошуку цього значення здійснюється за одним з наступних критеріїв:

- $\nabla^T P(x^{(0)}, r^{(k)})[\nabla^2 P(x^{(0)}, r^{(k)})]^{-1} \nabla P(x^{(0)}, r^{(k)}) < \varepsilon$ , тобто алгоритм зупиняється, коли

$$\frac{\partial P(x, r^{(k)}}{\partial x_i}(\Delta x_i) < \varepsilon.$$

- $\nabla^T P(x^{(0)}, r^{(k)})[\nabla^2 P(x^{(0)}, r^{(k)})]^{-1} \nabla P(x^{(0)}, r^{(k)}) < [P(x^{(k-1)}) - P(x^{(k)}]/5$ .

-

$$\left|\frac{\partial P(x, r^{(k)}}{\partial x_i}\right| < \varepsilon \quad i = \overline{1, n}.$$

• Вектор $x^{(k+1)}$ визначається за формулою

$$x^{(k+1)} = x^{(k)} + \lambda^{(k)} s^{(k)}. \tag{251}$$

Напрямок руху визначається:

- Якщо пошук точки $x^{(k)}$, що доставляє мінімум $P$ -функції, здійснюється за методом Ньютона

$$s^{(k)} = -[\nabla^2 P(x^{(k)}, r^{(k)})]^{-1} \nabla P(x^{(k)}, r^{(k)}). \tag{252}$$

- Якщо пошук здійснюється за методом ДФП

$$s^{(k)} = -\eta_0 \nabla P(x^{(k)}, r^{(k)}). \tag{253}$$

Довжина кроку визначається шляхом мінімізації модифікованої $P$ -функції.

• Після визначення напрямку пошуку здійснюється пошук за методом Фібоначчі з метою визначення положення мінімуму функції $P(x, r^{(k)})$ на відстані $\Delta x^{(k)}$ від $x^{(k)}$ у напрямку пошуку. Ця процедура виконується доти, доки матриця Гессе не стане додатно визначеною.

Отже, будується послідовність $x^{(k)}$, що мінімізує функцію $P(x^{(k)}, r^{(k)})$, а значить, й функцію мети $f(x)$ при обмеженнях у вигляді рівностей та нерівностей.

• Але у зв'язку з тим, що при наближенні до точки умовного мінімуму алгоритм МПБМ збігається повільно, оскільки поблизу межі процес мінімізації виконується повільно, то для прискорення цього процесу використовується засіб зменшення вагового коефіцієнта з метою отримання кількох мінімумів функції $P(x^{(k)}, r^{(0)})$ та відповідних до них точок $x^{(k)}$. Після цього значення складових вектору екстраполюються у вигляді поліномів за степенями $\sqrt{r}$ відповідно з формулами першого порядку

$$x(0) \approx \frac{\sqrt{c}x(r^{(k)}/c) - x(r^{(k)})}{\sqrt{c} - 1}, \tag{254}$$

або другого порядку

$$x(0) \approx \frac{\sqrt{c^3}x(r^{(k)}/c^2) - (c + \sqrt{c})x(r^{(k)}/c) + x(r^{(k)})}{(c-1)(\sqrt{c} - 1)}, \tag{255}$$

• Критерієм завершення оптимізаційного процесу пошуку є наступний.

$$r^{(k)} \sum_{i=m+1}^{p} \frac{1}{g_i[x(r^{(k)})]} \le \varepsilon. \tag{256}$$

Якщо ця нерівність задовольняється, процес закінчується, інакше переходимо до

наступного кроку.

• Значення вагового коефіцієнта зменшується. До речі, одним з можливих варіантів побудови послідовності $\{r^{(k)}\}$ є наступний: $r^{(0)} = 1$; $r^{(k)} = r^{(k-1)}/c$, де $c > 1$ (зазвичай приймають $c = 4$).

• За допомогою екстраполяційних формул (41) або (42)) виконується обчислення мінімуму функції $P(x, r^{(k)})$ для зменшеного значення $r^{(k)}$. Залежно від отриманого результату ітераційний процес продовжується (пункт 3), або завершується.

**Приклад**

Мінімізувати функцію $f(x) = 4x_1 - x_2^2 - 12$

за обмежень:

$$\begin{aligned} h_1(x) &= 25 - x_1^2 - x_2^2 = 0, \\ g_2(x) &= 10x_1 - x_1^2 + 10x_2 - x_2^2 - 34 \geq 0, \\ g_3(x) &= x_2 \geq 0, \\ g_4(x) &= x_2 \geq 0. \end{aligned}$$

Для наочності побудуємо графіки функцій $f(x)$, $h_1(x)$, $g_2(x)$ з урахуванням функцій -обмежень $g_3$ та $g_4$, що у площині $x_1, x_2$ обмежують значення інших функцій (параболи, кола та круга) першим квадрантом.

Оптимізаційний пошук почнемо з точки $x = [11]^T$. У цій точці $f(x) = -9$. Але ця точка є зовнішньою (відносно $g_2(x)$). Визначимо внутрішню точку $x$, що однак не є допустимою по відношенню до $h_1(x)$. Мінімізуємо функцію $-g_2(x)$ з обмеженнями $g_3(x) \geq 0$ та $g_4(x) \geq 0$. Побудуємо першу з допоміжних $P$ -функцій у вигляді

$$P' = -g_2(x) + r\sum_{i=3}^{4} \frac{1}{g_i(x)} = -(10x_1 - x_1^2 + 10x_2 - x_2^2 - 34) + (1)\left(\frac{1}{x_1} + \frac{1}{x_2}\right).$$

Маємо: $g_2(x^{(0)}) = 16$, $[1/x_1 + 1/x_2] = 2$; $P'(x^{(0)}) = 18$.

Мінімізуємо цю функцію за методом Ньютона. Визначимо

$$\begin{aligned} \frac{\partial P'}{\partial x_1} &= -10 + 2x_1 - \frac{1}{x_1^2}, \quad \frac{\partial P'}{\partial x_2} = -10 + 2x_2 - \frac{1}{x_2^2}, \\ \frac{\partial^2 P'}{\partial x_1^2} &= 2 + 2x_1^3, \quad \frac{\partial^2 P'}{\partial x_2^2} = 2 + \frac{2}{x_2^3}, \quad \frac{\partial^2 P'}{\partial x_1 \partial x_2} = 0. \end{aligned}$$

У точці $x = [11]^T$ маємо

$$\begin{aligned} s &= -H^{-1}\nabla P' = -\begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix}^{-1} \begin{bmatrix} -9 \\ -9 \end{bmatrix} = \begin{bmatrix} 2,25 \\ 2,25 \end{bmatrix}. \\ x^{(0)'} &= x^{(0)} + s = \begin{bmatrix} 3,25 \\ 3,25 \end{bmatrix}. \end{aligned}$$

Для цього вектора задовольняються усі обмежувальні умови у вигляді нерівностей, $f(x^{(0)'}) = -20,25$.

Тапер побудуємо головну $P$ -функцію

$$P = f(x) + \frac{h_1^2(x)}{\sqrt{r}} + r\left(\frac{1}{x_1} + \frac{1}{x_2} + \frac{1}{10x_1 - x_1^2 + 10x_2 - x_2^2 - 34}\right);$$

$$P(x^{(0)'}, 1) = 529,716.$$

У точці $x^{(0)'} = [3,25\ 3,25]^T$ похідні функції $P$ приймають значення

$$\frac{\partial P}{\partial x_1} = -46,505;\ \frac{\partial^2 P}{\partial x_1^2} = 69,084;\ \frac{\partial^2 P}{\partial x_1 \partial x_2} = 84,025;\ \frac{\partial P}{\partial x_2} = -57,006;\ \frac{\partial^2 P}{\partial x_2^2} = 69,084.$$

Обчислимо визначник матриці Гессе:

$$det\begin{bmatrix} 69,084 & 84,525 \\ 84,525 & 67,084 \end{bmatrix} < 0.$$

Значить, складові вектора, що визначають напрямок пошуку, направлені вздовж складових від'ємного градієнту функції $P$. Змінимо вектор

$$x = x^{(0)'} + \Delta x^{(0)} = x^{(0)'} - \nabla P = \begin{bmatrix} 49,755 \\ 60,256 \end{bmatrix}.$$

Далі процес мінімізації $P$-функції здійснюється за методом ``золотого перетину".

Обчислимо значення $P(x^{(0)}) = 6,17$ та $P(x^{'(0)}) = 0,284 \cdot 10^7$.

Згідно з алгоритмом методу "золотого перетину" необхідно визначити третю точку, у якій виконується умова $P(x_3^{(0)}) > P(x^{'(0)}) = P(x_2^{(0)})$.

Такою точкою є $x_3^{(0)} = (x_1^{(0)} + x_2^{(0)})/2 = [24,877\ \ 30,128]$, в якій $P(x_3^{(0)}) = 0,37 \cdot 10^8$. Визначимо два дроби Фібоначчі: $F_1 = (3 - \sqrt{5})/2$ та $F_1 = (\sqrt{5} - 1)/2$.

Позначимо $\Delta^{(0)} = x_3^{(0)} - x_1^{(0)} = [21,627\ \ 26,878]$.

Визначимо $u_1^{(0)} = x_1^{(0)} + F_1\Delta^{(0)} = [21,014\ 25,024]$,

$u_2^{(0)} = x_1^{(0)} + F_2\Delta^{(0)} = [31,992\ \ 38,481]$.

Маємо: $P(u_1^{(0)}) = 0,1087 \cdot 10^7 < P(u_2^{(0)}) = 0,6146 \cdot 10^7$.

Тоді $\Delta^{(1)} = u_2^{(0)} - x_1^{(0)} = [28,742\ \ 35,231]$

Вектор $x^{(1)} = [3,25\ \ 3,25\ \ 31,992\ \ 38,481]$; $\delta^{(1)} = 35,23$.

Оскільки $\delta > \varepsilon = 0,01$, виконуємо другу ітерацію:

$u^{(1)} = [14,228\ \ 16,707\ \ 21,014\ \ 25,024]$;

$P(u_1^{(1)}) = 0,2082 \cdot 10^6 < P(u_2^{(1)}) = 0,1087 \cdot 10^7$;

$x^{(2)} = [3,25\ \ 3,25\ \ 21,014\ \ 25,024]$; $\delta^{(1)} = 21,77$.

$\Delta^{(2)} = [17,764\ \ 21,774]$; $u^{(2)} = [10,035\ \ 11,567\ \ 14,228\ \ 16,707]$;

$x^{(2)} = [3,25\ \ 3,25\ \ 14,228\ \ 16,707]$; $\delta^{(1)} = 13,46$.

Далі продовжуємо обчислення доки не справдиться нерівність $\delta < \varepsilon_0 = 0,01$.

Після 17 ітерацій отримуємо:

$x^{(18)} = [3,5128\ \ 3,5721\ \ 3,5209\ \ 3,582]$; $\delta^{(18)} = 0,009866$.

Таким чином, внаслідок реалізації методу ``золотого перетину" отримана точка мінімуму $x^{(0)} = [3,5168\ \ 3,5771]$.

Обчислимо значення основного функціоналу ((44)) $P(x^{(0)})$ та його похідних першого та другого порядку. Маємо: $\partial P(x^{(0)})/\partial x = [6{,}119 \;\; -4{,}993]$.

$$\frac{\partial^2 P(x^{(0)})}{\partial x^2} = \begin{bmatrix} 99{,}63 & 100{,}6 \\ 100{,}6 & 101 \end{bmatrix}.$$

Визначник цієї матриці знову від'ємний. Отже ще раз застосовуємо метод ``золотого перетину''. Після 14 ітерацій отримуємо: $x^{(0)} = [2{,}1373 \;\; 4{,}7021]$.

$$\frac{\partial P(x^{(0)})}{\partial x} = \begin{bmatrix} 18{,}01 \\ 22{,}05 \end{bmatrix}; \quad \frac{\partial^2 P(x^{(0)})}{\partial x^2} = \begin{bmatrix} 43{,}64 & 80{,}42 \\ 80{,}42 & 181{,}6 \end{bmatrix}.$$

Оскільки визначник цієї матриці додатній, подальші обчислення виконуємо згідно з формулою Ньютона (39), тобто, переходимо до другого етапу реалізації методу послідовної безумовної мінімізації задачі.

$x^{(1)} = [1{,}104 \;\; 5{,}0352]$; $\delta^{(1)} = 1{,}027 > 0{,}0001$.

Після п'яти ітерацій (параметр $r$ -- ваговий коефіцієнт функції $P(x)$) отримуємо точку локального мінімуму

$x^{(5)} = [1{,}1505 \;\; 4{,}9178]$; $\delta^{(1)} = 0{,}66 \cdot 10^{-4} < 0{,}0001$.

Тепер обчислимо у цій точці значення функції мети $f(x)$ та $h_1(x)$.

$f(x) = -31{,}58$; $h_1(x) = -0{,}5082$.

Оскільки значення функції $h_1$ у точці мінімуму не дорівнює нулю (або ж не менше потрібної точності), зменшуємо значення вагового коефіцієнта $r = 0{,}5$, та повторюємо пошук мінімуму за формулою Ньютона. Й цього разу довелось виконати п'ять ітерацій для досягнення точності $\varepsilon = 0{,}0001$. Взагалі кажучи, кількість ітерацій за методом Ньютона пропорційна бажаній точності результату, тобто, для досягнення точності $0{,}01$ у більшості випадків досить двох ітерацій методу Ньютона.

Отже маємо: $x^{(10)} = [1{,}1045 \;\; 4{,}9132]$ ; $f(x) = -31{,}72$ ; $h_1(x) = -0{,}359$ . Подальші обчислення при зменшенні на кожному кроці параметра $r = r/2$ дозволили отримати наступний розв'язок задачі з обмеженням у вигляді рівності та трьох обмежень у вигляді нерівностей:

$x = [1{,}1035 \;\; 4{,}8991]$; $f(x) = -31{,}99$; $h_1(x) = -0{,}007934$ при $r = 0{,}244 \cdot 10^{-3}$.

Траєкторію руху до точки мінімуму схематично відображено на рис. 4.1..

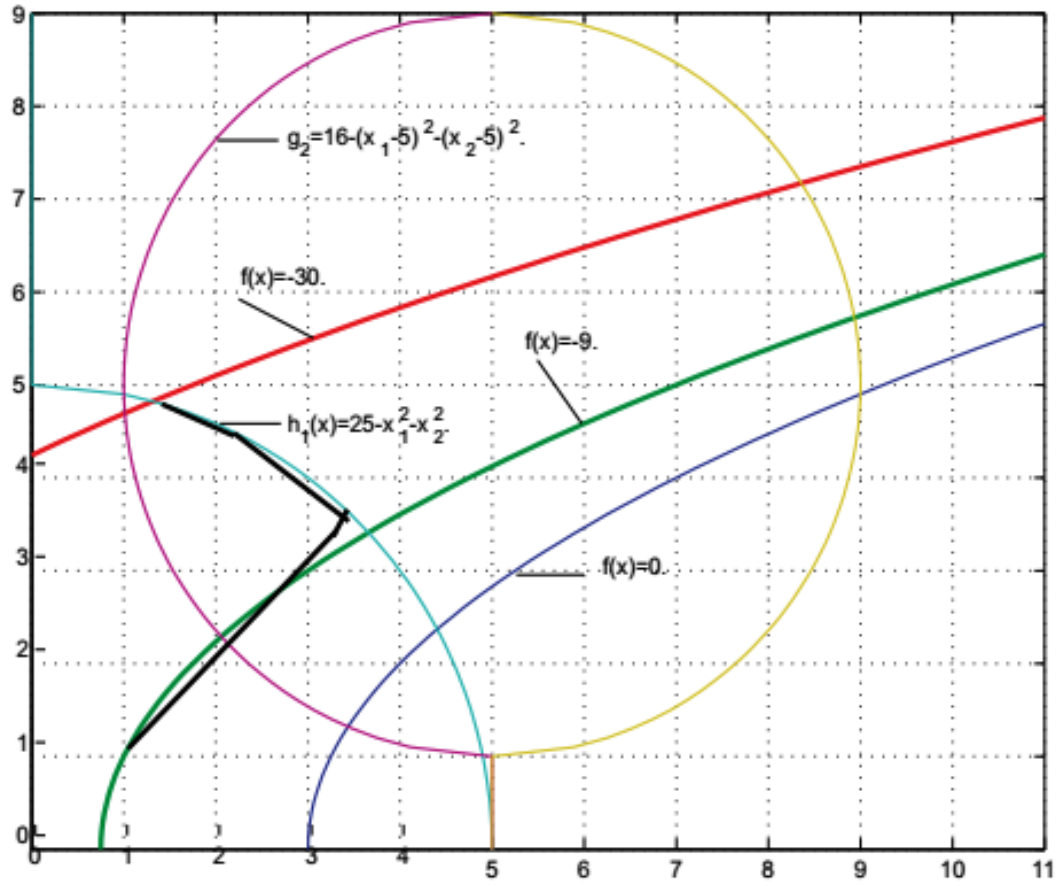


Рис. 4.1. Оптимізація за методом МПБМ

# Розділ 5 Динамічне програмування

На відміну від принципу максимуму, який формулюється таким чином, що виявляється орієнтованим, перш за все, на визначення оптимального управління у вигляді оптимальної програми, динамічне програмування надає можливість визначати оптимальне управління тільки у формі синтезуючої функції.

Динамічне програмування добре обгрунтовано для дискретних систем і процесів. Це пов'язано із тим, що при виводі функціонального рівняння Беллмана треба робити припущення, безпосередня перевірка якого за рівняннями руху та функціоналу якості неможлива. Це можна зробити тільки після розв'язання рівняння Беллмана. Крім того, функціональне рівняння Беллмана для неперервних процесів являє собою диференційне рівняння із частинними похідними.

Якщо мати на увазі не тільки задачі оптимального управління, динамічному програмуванню властива універсальність й його можна застосовувати для вирішення широкого кола задач оптимізації. Будемо розглядати основний зміст динамічного програмування як методу оптимізації.

Розглянемо деяку систему $S$ (економічну, соціальну, фізичну тощо), стани якої змінюються у дискретні моменти часу $t_0, t_1, \ldots, t_m$, утворюючи деякий процес, що складається із $m$ етапів: $[t_0, t_1]$, $\ldots, [t_{m-1}, t_m]$. Стани $S_i$ системи $S$ у момент $t_i$, $i = \overline{1, m}$, опишемо деяким вектором

$$x^{(i)} = [x_1^{(i)}(t_i), x_2^{(i)}(t_i), \ldots, x_n^{(i)}(t_i)]^T \in R^n.$$

Координати цього вектора називатимемо фазовими траєкторіями (або фазовими координатами). Сукупність станів характеризує процес переходу від першого стану $x^{(0)}$ до останнього $x^{(m)}$.

Наприклад, для системи, що описується рівнянням другого порядку

$$\frac{d^2x(t)}{dt^2} = f(x,t),$$

стан системи визначається змінними стану $x$ та $dx/dt$, тобто визначається функціями, що визначають рух точки $x$ та її швидкість. Графіки цих функцій визначають так звані фазові траєкторії системи.

Передбачається, що на кожному етапі процес визначається як власною динамікою системи (станами), так і зовнішніми впливами

$$u^{(i)} = [u_1^{(i)}(t_i), u_2^{(i)}(t_i), \ldots, u_n^{(i)}(t_i)]^T \in R^r; i = \overline{1, m}.$$

Назвемо ці впливи *керуваннями*. У результаті $i$-го етапу процес переходить у стан $x^{(i)}$, який визначається тільки станом $x^{(i-1)}$ та вибраним на цьому етапі керуванням $u^{(i)}$ і не залежить від ``передісторії'' процесу до $i$-го етапу. Зв'язок між станами можна описати рівняннями стану

$$x^{(i)} = f^{(i)}\left(x^{(i-1)}, x^{(i)}\right), \qquad (1)$$

де $f^{(i)}(x,u)$, $i = \overline{1, m}$ -- деякі задані функції. Будемо вважати, що ефективність $i$-го стану оцінюється функцією мети $J^{(i)}(x^{(i-1)}, x^{(i)})$, а ефективність усього процесу -- адитивною функцією мети вигляду

$$J\left(x^{(0)},\ldots,x^{(m-1)};u^{(1)},\ldots,u^{(m)}\right)=\sum_{i=1}^{m}J^{(i)}\left(x^{(i-1)},u^{(i)}\right). \quad (2)$$

На фазову траєкторію і вибір керування можуть бути накладені обмеження вигляду

$$x^{(i)}\in X^{(i)}, i=\overline{1,m-1}, \quad (3)$$

$$u^{(i)}\in U^{(i)}(x^{(i-1)}), i=\overline{1,m}, \quad (4)$$

де $X^{(i)}\subset R^n$, $U^{(i-1)}\subset R^r$ -- задані множини. Фукнції (3) та (4), що не виходять за межі множин, називають допустими станами та керуваннями.

Окремо виділимо обмеження на початковий та кінцевий стани процесу:

$$x^{(0)}\in X^{(0)},\ x^{(m)}\in X^{(m)}. \quad (5)$$

У більшості випадків множини станів $X^{(0)}$ і $X^{(m)}$ містять тільки по одній точці, тобто початкову і кінцеву траєкторії.

*Задача оптимізації* (або оптимального керування) описаним процесом ставиться так: *серед усіх можливих (допустимих) керувань знайти таке керування* $\{u^{(1)*},\ldots,\ u^{(m)*}\}$*, яке за наявності умов (3), (4) переводить систему* $S$ *зі стану* $x^{(0)}$ *у стан* $x^{(m)}$ *і для якого цільова функція (3) досягає свого екстремального значення*.

Керування $\{u^{(1)*},\ldots,u^{(m)*}\}$ та відповідна до нього траєкторія $\{x^{(0)*},\ldots,\ x^{(m-1)*}\}$ називаються *оптимальними*.

Математична модель цієї задачі на максимум має вигляд:

$$\left.\begin{array}{ll}\max\limits_{u} & J(x^{(0)},\ldots,x^{(m-1)};u^{(1)},\ldots,u^{(m)}),\\ x^{(i)} & =f^{(i)}\left(x^{(i-1)},u^{(i)}\right),\ i=\overline{1,m};\\ u^{(i)} & \in U^{(i)}(x^{(i-1)}),\ i=\overline{1,m},\\ x^{(0)} & \in X^{(0)}.\end{array}\right\} \quad (6)$$

Ця задача описує як дискретну, так і неперервну модель процесу оптимізації.

Пошук екстремуму функції мети $J$ здійснюється поетапно, починаючи з останнього стану та керування.

Це стає можливим, коли функція мети є адитивна функція.

## Адитивні функції і задачі

*Визначення*. Функція $f(x^{(0)},\ldots,x^{(N)})$ має назву адитивної, якщо вона має вигляд

$$f(x^{(0)},\ldots,x^{(N)})=\sum_{i=0}^{N-1}f_i(x^{(i)},x^{(i+1)}), \quad (7)$$

де $x^{(k)}$ -- вектори розміром $n_k$.

Це означає, що функція мети складається із суми функцій, що визначають стани системи на двох сусідніх значеннях цих станів. Для статичних процесів відома аналогічна теорема Колмогорова про подання функції багатьох змінних у вигляді суми адитивних функцій.

Якщо обмеження мають вигляд

$$x^{(i)} \in G_i,\ i = \overline{0, N}, \tag{8}$$

задача пошуку мінімуму функції (7) також називається *адитивною.*

Адитивні задачі мають місце у багатьох галузях практичної діяльності людини. Вони мають наочну геометричну інтерпретацію, сутність якої полягає у такому. За припущення, що всі $n_k = n$, запровадимо у розгляд $(N+1)$-вимірний простір пар $\{x, T\}$, в якому побудовано гіперплощини $\Sigma$, що відповідають значенням $t = i$, $i = \overline{1, N}$. Тоді будь-яку сукупність векторів можна ототожнити з ламаною, що проходить у $(N+1)$-вимірному просторі через точки $\{x^{(0)}, 0\} \in \Sigma_0$, …, $\{x^{(N)}, N\} \in \Sigma_N$. Її загальну довжину визначимо як суму довжин її ланцюгів, а довжину ланцюга, що сполучає точки $\{x^{(i)}, i\}$, $\{x^{(i+1)}, i+1\}$, будемо вимірювати величиною $f_i\left(x^{(i)}, x^{(i+1)}\right)$. У цьому випадку довжина усієї ламаної є $f\left(x^{(0)}, \ldots, x^{(N)}\right)$.

Тепер початкову задачу можна сформулювати так: серед усіх допустимих $x^{(i)} \in G_i$) ламаних, що з'єднують гіперплощини $\Sigma_0$ та $\Sigma_N$, знайти ту, довжина якої є мінімальна.

У подальшому вектори $x^{(i)}$ будемо ототожнювати із точками $\{x_i, i\} \in \Sigma_i$. Множину усіх ламаних позначимо через $\Omega$.

Розглянемо довільну точку $x_1 \in \Sigma_1$. Найкоротший відрізок, що з'єднує цю точку із гіперплощиною $\Sigma_0$, має довжину

$$l(x_1) = \min_{x_0 \in G_0} f_0(x_0, x_1). \tag{9}$$

Оскільки

$$\min_{x_0 \in G_0} f(x_0, \ldots, x_N) = l(x_1) + \sum_{i=1}^{N-1} f_i(x_i, x_{i+1}),$$

будь-яка ламана, що не містить відрізку $l(x_1)$, не може бути претендентом на те, щоб вона вважалася розв'язанням задачі мінімізації. Ці ламані утворюють множину $\omega_0$, яка й відкидається на першому кроці: $\Omega_1 \setminus \omega_0$. Якщо продовжити ці міркування, отримаємо найкоротшу ламану, що з'єднує точку $x_{i+1}$ із гіперплощиною $\Sigma_0$, якою буде

$$l(x_{i+1}) = \min_{x_i \in G_i} [l(x_i) + f_i(x_i, x_{i+1})]. \tag{10}$$

Ця формула являє собою рекурентне співвідношення, що описує багатокроковий процес пошуку розв'язання, яке приводить задачу пошуку мінімуму функції $n$ змінних до послідовного пошуку мінімуму $n$ функцій однієї змінної.

Структура адитивних задач є зручна і для застосування градієнтного спуску, оскільки

$$\frac{\partial f}{\partial x_i} = \frac{\partial f_i}{\partial x_i} + \frac{\partial f_{i-1}}{\partial x_i}.$$

Будемо говорити, що процес задовольняє принцип оптимальності, якщо будь- яка ділянка оптимальної траєкторії є оптимальна траєкторія.

## 5.1 Дискретні керуючі системи

Розглянемо динамічний процес, що протікає у дискретному часі. Позначимо стан

процесу у момент $t_i$ через $x(t_i)$, де $x$ -- $n$-вимірний вектор і припустимо,що він описується рівнянням

$$x(t_{i+1}) = F_i[x(t_{i+1}), u(t_i)], \ i = \overline{0, N-1}. \tag{11}$$

Початковий стан вважаємо заданим

$$x(t_0) = x_0. \tag{12}$$

Задача полягає у пошуку мінімуму функції вигляду

$$J = J(x_1, \ldots, x_N; u_1, \ldots, u_N) \tag{13}$$

із обмеженнями

$$x_i = x(t_i) \in X_i, \ i = \overline{0, N}, \tag{14}$$

$$u_i = u(t_i) \in U_i, \ i = \overline{0, N-1}. \tag{15}$$

Для того, щоб задача відшукання фазової траєкторії $x_i$ та керування $u_i$, що реалізує мінімум функціоналу $J_i$ приводилася до адитивного вигляду, дискретна система повинна мати низку властивостей. Насамперед, функціонал має бути адитивним, тобто

$$J = \sum_{i=0}^{N-1} f_i(x_i, x_{i+1}, u_i). \tag{16}$$

Крім того, система має допускати існування так званої *елементарної операції*, яка буде визначеною, якщо парі точок $x_i, x_{i+1}$ можна співставити у відповідність керування $u_i$, що переводить систему за один такт із стану $x_i$ у стан $x_{i+1}$. Тобто

$$u_i = B(x_i, x_{i+1}). \tag{17}$$

Тоді функція (16) прийме вигляд

$$J = \sum_{i=0}^{N-1} \tilde{f}_i(x_i, x_{i+1}), \tag{18}$$

де

$$\tilde{f}_i(x_i, x_{i+1}) = f_i(x_i, x_{i+1}; B(x_i, x_{i+1})).$$

Для того, щоб елементарна операція (17) могла бути побудована, необхідно, щоб система (11) мала розв'язання відносно $u_i$. Для цього розмірність вектора $u_i$ має бути не менша за розмірність $x_i$.

Отже, якщо керування можна виключити із функціоналу, отримаємо адитивну задачу.

Інший напрямок дослідження дискретних керованих систем пов'язаний із виключенням фазових змінних.

Розглянемо для визначеності задачу із функцією мети

$$J = \sum_{i=1}^{N} F_i(x_i, u_{i-1}), \quad F_N = F_N(x_N) \tag{19}$$

та кінцево-різницевими рівняннями

$$x_i = f_i(x_i, u_i), \ i = \overline{0, N-1}. \tag{20}$$

Нехай, крім того, вектори $x_i, u_i$ задовольняють умови (13)--(15). Оскільки початковий стан $x_0$ заданий, можна вважати, що $f_0 = f_0(u_0)$. Тоді з урахуванням рівняння (20) можна виключити вектори $x_i$ із функціоналу (19):

$$J = F_1(u_0,\, f_0(u_0) + F_2(u_1,\, f_1(u_1,\, f_0(u_0))) +$$
$$\ldots + F_N(\, f_{N-1}(u_{N-1},\, f_{N-1}(u_{N-2},\, f_{N-2}\ldots))\ldots). \qquad (21)$$

Функції такого вигляду звуться *функціями з послідовним виключенням змінних.*

### 5.1.1 Загальна схема динамічного програмування

Розглянемо останній, $N$ -й крок процесу, що описується системою рівнянь (20) і припустимо, що система перебуває при $t = N-1$ у деякому фіксованому стані $x_{N-1}$. Позначимо

$$J_k = \sum_{i=1}^{k} F_i(x_i, u_{i-1}). \qquad (22)$$

Тоді

$$J = J_{N-1} + F_N(x_N).$$

Із урахуванням (186) подамо $J$ у вигляді

$$J = J_{N-1} + F_N(\, f_{N-1}(x_{N-1}, u_{N-1}).$$

Якщо система уже перебуває у стані $x_{N-1}$, треба вибрати $u_{N-1}$ так, щоб мінімізувати функцію

$$F_N(\, f_{N-1}(x_{N-1}, u_{N-1})).$$

Позначимо

$$S_{N-1}(x_{N-1}) = \min_{u_{N-1} \in U_{N-1}} F_N(\, f_{N-1}(x_{N-1}, u_{N-1})). \qquad (23)$$

Ця рівність визначає деяку функцію розташування $x_{N-1}$. Величина

$$J = J_{N-1} + S_{N-1}(x_{N-1})$$

визначає те мінімальне значення функцій $J$, яке може бути досягнуто, якщо траєкторія системи пройшла через точки $x_0, x_1, \ldots, x_{N-1}$. У процесі обчислення величини $S_{N-1}(x_{N-1})$ відшукується керування $u_{N-1}$, що доставляє найменше значення функції (що задана)

$$F_N(\, f_{N-1}(x_{N-1}, u_{N-1})).$$

Отже, одночасно із скалярною функцією $S_{N-1}(x_{N-1})$ визначається й вектор- функція

$$u_{N-1} = \varphi_{N-1}(x_{N-1}), \qquad (24)$$

яка ставить у відповідність кожному фазовому стану $x_{N-1}$ керування $u_{N-1}$.

Фактично уже сформульовано правило відкидання не конкурентно- спроможних варіантів.

Розглянемо тепер другий крок. Функцію $J$ можна тепер записати так:

$$J = J_{N-2} + F_{N-1}(x_{N-1}, u_{N-1}) + S_{N-1}$$
$$= J_{N-2} + F_{N-2}(\, f_{N-2}(x_{N-2}, u_{N-2}) + S_{N-1}(\, f_{N-2}(x_{N-2}, u_{N-2}). \qquad (25)$$

Якщо зафіксувати точку $x_{N-2}$, найменше значення функції $J$ визначиться тільки керуванням $u_{N-2}$. Оскільки $u_{N-2}$ входить тільки у дві останні складові в (191), то якщо знайти функцію

$$S_{N-2}(x_{N-2}) = \min_{u_{N-2}\in U_{N-2}} \{F_{N-1}(f_{N-2}(x_{N-2},u_{N-2})) + S_{N-1}(f_{N-2}(x_{N-2},u_{N-2}))\},$$

отримаємо мінімальне значення функції $J$:

$$J = J_{N-2} + S_{N-2}(x_{N-2}),$$

яке може бути досягнуто, коли траєкторія системи пройшла через точки $x_0$, $x_1$, …, $x_{N-2}$. Одночасно із $S_{N-2}(x_{N-2})$ визначається також керування

$$u_{N-2} = \varphi_{N-2}(x_{N-2}).$$

Отже, якщо нам відомо, що у ``момент часу'' $t = N-2$ система перебуває у стані $x_{N-2}$, для отримання найменшого значення функції $J$ керування $u_{N-2}$, $u_{N-1}$ на двох останніх кроках слід обрати відповідно до формул

$$u_{N-2} = \varphi_{N-2}(x_{N-2});$$

$$u_{N-1} = \varphi_{N-1}(x_{N-1}) = \varphi_{N-1}[f_{N-1}(x_{N-2},\varphi_{N-2}(x_{N-2}))].$$

На цьому кроці ми виключаємо із множини $\Omega_1$ траєкторій, що залишилися, множину всіх траєкторій, для яких

$$u_{N-2} \neq \varphi_{N-2}(x_{N-2}).$$

Якщо процес продовжити, на кожному кроці визначимо функцію

$$S_{i-1}(x_{i-1}) = \min_{u_{i-1}\in U_{i-1}} \{F_i(f_{i-1}(x_{i-1},u_{i-1})) \qquad (26)$$

яка стану $x_{i-1}$ системи за формулою

$$J = J_{i-1} + S_{i-1}(x_{i-1})$$

становить у відповідність те мінімальне значення $J$, яке може бути досягнуто, якщо траєкторія системи пройшла через точки $x_0, x_1, \ldots, x_{i-1}$. Одночасно отримаємо функцію

$$u_{i-1} = \varphi_{i-1}(x_{i-1}).$$

Нарешті припустимо, що визначено функції $S_1(x_1)$ та $u_1 = \varphi_1(x_1)$; тоді

$$J = J_0 + S_1(x_1) = F_0(x_0) + S_1(x_1),$$

тож нам залишилося тільки визначити величину $u_0$ та мінімальне значення $S_0$ при заданому початковому стані $x_0$:

$$S_0 = \min_{x_0\in U_0} \{F_0(u_0) + S_1(x_0,u_0).$$

$$u_0 = \varphi_0(x_0).$$

Отже, задачу вирішено. Число $S_0$ -- це мінімальне значення функції $J$ на множині

$$U = U_0 \times U_1 \times \cdots \times U_{N-1}.$$

Для реалізації цього значення $S_0$ треба побудувати послідовність векторів

$$u_0 = \varphi_0(x_0).$$

$$u_1 = \varphi_1(x_1) = \varphi_1[f_0(x_0,\varphi_0(u_0));$$

$$u_2 = \varphi_2(x_2) = \varphi_2[f_1(x_1,u_1)]$$

$$= \varphi_2[f_1(f_0(x_0,\varphi_0(x_0),\varphi_1(f_0(x_0),\varphi_0(x_0)))];\ldots$$

Рівняння (26), що визначає керування $u_{i-1}$, має назву *рівняння Беллмана.*

### 5.1.2 Числова реалізація

Для побудови числової схеми у просторі $(x,t)$ побудуємо сітку із кроком $\Delta x$. Вузли цієї сітки позначимо через $P_k(i)$. На першому кроці процесу треба знайти функцію

$$S_{N-1}(P_k(N-1)) = \min_j F_N(f_{N-1}(P_k(N-1), u_{N-1}(j))).$$

На цьому ж кроці ми отримаємо також

$$u_{N-1}(k) = \varphi_{N-1}(P_k(N-1)).$$

На наступному кроці ми маємо побудувати таблицю для функції

$$S_{N-2}(P_k(N-2)) = \min_j \{F_{N-1}(f_{N-2}(P_k(N-2), u_{N-2}(j)))$$
$$+S_{N-1}(f_{N-2}(P_k(N-2), u_{N-2}(j)))\}.$$

Але, оскільки функція $S_{N-1}$ задана таблично на сітці вузлів $P_i(N-1)$, серед цих вузлів може не виявитися вузла такого, щоб його координати визначалися компонентами вектора $(f_{N-2}(P_k(N-2), u_{N-2}(j))$. Тобто точка $x_{N-1}$, яка відповідає керуванню $u_{N-2}(j)$, може не збігатися із жодним вузлом у гіперплощині $\Sigma_{N-1}$, а, значить, вона відсутня у таблиці значень функції $S_{N-1}$.

Перехід із заданого стану $x_{N-2}$ у заданий стан $x_{N-1}$ можливий тільки тоді, коли існує керування $u_{N-2}$, яке є корінь векторного рівняння

$$x_{N-1} = f_{N-1}(x_{N-2}, u_{N-2}). \qquad (27)$$

Для цього необхідно, щоб розмірність вектора $u$ була не менше за розмірність вектору $x$. У цьому випадку за фіксованого $x_{N-2}$ формула (27) дає деяке відображення $U_{N-2}$ на $\Sigma_{N-2}$. Позначимо його через $Q(U_{N-2}, x_{N-2})$. Якщо деякий вузол $P_j(N-1) \in Q(U_{N-2}, x_{N-2})$, говорять, що він є досяжний із точки $x_{N-2}$.

Елементарною операцією (стосовно даного випадку) називають процедуру відшукання дійсного кореня рівняння (27), тобто функції $u_{N-2}$, що залежить від $x_{N-1}$ та $x_{N-2}$:

$$u_{N-2} = \varphi^*_{N-2}(x_{N-1}, x_{N-2}).$$

Тепер можна подати функцію $S_{N-2}$ у такій формі:

$$S_{N-2}(P_k(N-2)) = \min_j \{F_{N-1}[f_{N-2}(P_k(N-2), \varphi^*_{N-2}(P_j(N-1), P_k(N-2))]$$
$$+S_{N-1}[f_{N-2}(P_k(N-2), \varphi^*_{N-2}(P_j(N-1), P_k(N-2))]\}.$$

Для побудови таблиці функції $S_{N-2}(P_k(N-2))$ досить перебрати значення вузлів $P_j(N-1)$. Відмітимо, що досить заповнити таблицю керування $u_{N-2}(P_k(N-2))$.

Якщо розмірність вектора $u$ менша за розмірність вектору $x$, ситуація суттєво ускладнюється, оскільки розмірність множини досяжності у загальному випадку виявляється меншою за розмірність простору $\Sigma_{N-1}$. У цьому випадку треба модифікувати алгоритм побудови.

Кожний із вузлів $P_j(i)$ оточимо деякою множиною $R_j(i)$, тобто вузол $P_j(i)$ є центр

деякого паралелепіпеду з довжиною ребра $2\varepsilon$. Будемо відносити до числа вузлів, що є недосяжні із точки $x_{i-1}$, усі вузли, в околі яких $R_j(i)$ мають із множиною досяжності спільні точки.

Отже, нехай функцію $S_i(x_i)$ уже побудовано. Це означає, що ми маємо таблицю значень $S_i(P_j(i))$ та керування $u_i(x_i)$. Запишемо функції $S_i(x_i)$ та $F_i(x_i, u_i)$ у такому вигляді:

$$S_i(x_i) = S_i[\, f_{i-1}(x_{i-1}, u_{i-1}(j_k))];$$
$$F_i(P_k(i)) = F_i[\, f_{i-1}(x_{i-1}, u_{i-1}(j_k))]. \qquad (28)$$

За цими таблицями ми легко будуємо таблицю для функції

$$S_{i-1}(x_{i-1}) = \min_k \{F_i[\, f_{i-1}(x_{i-1}, u_{i-1}(j_k))] + S_i[\, f_{i-1}(x_{i-1}, u_{i-1}(j_k))]\}. \qquad (29)$$

Операція (29) кожному вузлу $P_k(i-1)$, що належить площині $\Sigma_{i-1}$, співставляє у відповідність єдине значення керування $u_{i-1}(j_k)$. Цей процес надає можливість отримати деяку послідовність керувань $u_0, u_1, \ldots, u_{N-1}$ та за допомогою формули (20) побудувати фазову траєкторію $x_0, x_1, \ldots, x_N$, й отже, обчислити мінімальне значення $J$.

Викладена процедура надає можливість обчислити мінімальне значення $J$ тим точніше, чим на густішій сітці виконуються обчислення. У зв'язку з цим для розв'язання подібних задач виникає потреба у використанні ітеративних методів.

### 5.1.3 Приклади

**Приклад 1.** У визначений момент часу на підприємстві встановлено нове обладнання. Залежність продуктивності цього обладнання від часу його використання, а також залежність витрат на утримання і ремонт обладнання відомі і наведені у табл. 5.1. $R(\tau)$ -- річний випуск продукції; $Z(\tau)$ -- річні витрати, що пов'язані із утриманням та ремонтом обладнання.

Табл. 5.1 Залежність продуктивності цього обладнання від часу

| Роки | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| $R(\tau)$ (у.о.) | 80 | 75 | 65 | 60 | 60 | 55 |
| $Z(\tau)$ (у.о.) | 20 | 25 | 30 | 35 | 40 | 55 |

Відомо, що витрати, пов'язані із придбанням і встановленням обладнання, що замінюється, списується.

Треба скласти такий план заміни обладнання упродовж п'яти років, за якого загальний прибуток за даний період часу був би максимальним.

*Розв'язання.*

Це завдання можна розглядати як завдання динамічного програмування, в якому в якості системи $S$ є обладнання. Стани цієї системи визначаються фактичним часом використання обладнання (його віком) $\tau$, тобто описуються одним параметром -- $\tau$. Керуваннями є рішення про заміну або збереження обладнання, які приймаються на початку кожного року.

Позначимо $u_1$ -- рішення про збереження обладнання, а $u_2$ -- рішення про заміну обладнання. Тоді завдання полягатиме у відшуканні такої стратегії керування, яка визначається рішеннями, що приймаються до початку кожного року, за якої загальний прибуток підприємства за п'ять років буде максимальний.

Отже, сформульоване вхідне завдання у термінах завдання динамічного програмування. Воно характеризується властивостями адитивності і відсутністю післядії. Розв'язання задачі можна знайти за допомогою розв'язання задачі динамічного програмування, що реалізується у два етапи. На першому етапі, рухаючись від початку п'ятого року до початку першого року для кожного допустимого стану обладнання знайдемо умовне оптимальне керування (рішення),а на другому, рухаючись від початку першого року до почату п'ятого року із умовних оптимальних рішень для кожного року оберемо оптимальний план заміни обладнання.

Для визначення умовних оптимальних рішень спочатку необхідно скласти функціональне рівняння Беллмана. Оскільки ми припустили, що до початку $k$-го року ( $k=\overline{1,5}$ ) може прийматися тільки одне з двох рішень -- замінювати чи не замінювати обладнання, прибуток підприємства за $k$-й рік складає

$$F_k(\tau^{(k)},u_k)=\begin{cases} R(\tau^{(k)})-Z(\tau^{(k)}), & при \quad u_1, \\ R(\tau^{(k)}=0)-Z(\tau^{(k)}=0)-C_n, & при \quad u_2, \end{cases}$$

де $\tau^{(k)}$ -- вік обладнання на початок $k$-го року; $u_k$ -- керування, яке реалізується на початок $k$-го року; $C_n$ -- коштовність нового обладнання.

Отже, у даному випадку рівняння Беллмана має вигляд

$$F_k(\tau^{(k)})=\max_{\tau}\begin{cases} R(\tau^{(k)}) & -Z(\tau^{(k)})+F_{k+1}(\tau^{(k+1)}), \\ R(\tau^{(k)}= & 0)-Z(\tau^{(k)}=0)-C_n+F_{k+1}(\tau^{(k+1)}=1), \end{cases} \qquad (30)$$

Почнемо із визначення умовно оптимального керування (рішення) для останнього (п'ятого) року, для чого знаходимо множину допустимих станів обладнання на початок даного року.

Використовуючи рівняння Беллмана, переходимо до розв'язання початкової задачі. Почнемо з визначення умовного оптимального керування для останнього (п'ятого) року, для чого знаходимо множину допустимих станів обладнання на початок даного року. Оскільки у початковий момент у наявності є нове обладнання ( $\tau^{(1)}=0$ ), вік обладнання на початок п'ятого року може складати 1, 2, 3 і 4 роки. Тому допустимі стани системи на даний період часу такі: $\tau^{(5)}=1$, $\tau^{(5)}=2$, $\tau^{(5)}=3$, $\tau^{(5)}=4$. Для кожного з цих станів знайдемо умовно оптимальний розв'язок і відповідне до нього значення функцій $F_5(\tau^{(5)})$ . Із (**Ошибка! Источник ссылки не найден.**) і співвідношення $F_5(\tau^{(k+1)})=0$ (розглядається останній рік розрахункового періоду). отримаємо

$$F_5(\tau^{(5)})=\max_{\tau}\begin{cases} R(\tau^{(5)}) & -Z(\tau^{(5)}), \\ R(\tau^{(5)}= & 0)-Z(\tau^{(5)}=0)-C_n, \end{cases} \qquad (31)$$

Із урахуванням $\tau^{(5)}=1$, а також даних, що наведено у табл. 5.1, отримуємо:

$$F_5(\tau_1^{(5)}) = \max\begin{cases} R(\tau^{(5)}=1) & -Z(\tau^{(5)}=1), \\ R(\tau^{(5)}=0) & -Z(\tau^{(5)}=0)-C_n, \end{cases} = \max\begin{cases} 75 & -25, \\ 80 & -20-40, \end{cases} = 50,$$

$$u^o = u_1.$$

Виконаємо аналогічні обчислення для інших допустимих станів обладнання на початок п'ятого року:

$$F_5(\tau_2^{(5)}) = \max\begin{cases} 65 & -30, \\ 80 & -20-40, \end{cases} = 35,\ u^o = u_1.$$

$$F_5(\tau_3^{(5)}) = \max\begin{cases} 60 & -35, \\ 80 & -20-40, \end{cases} = 25,\ u^o = u_1.$$

$$F_5(\tau_4^{(5)}) = \max\begin{cases} 65 & -45, \\ 80 & -20-40, \end{cases} = 20,\ u^o = u_2.$$

Табл.5.2 Умовно оптимальний розв'язок для $\tau^{(5)}$

| Вік обл. $\tau^{(5)}$ | $F_5(\tau^{(5)})$ (у.о.) | $u^o$ |
|---|---|---|
| 1 | 50 | $u_1$ |
| 2 | 35 | $u_1$ |
| 3 | 25 | $u_1$ |
| 4 | 20 | $u_2$ |

Розглянемо тепер можливі стани на початок 4-го року. Вочевидь, допустимими станами є $\tau_1^{(4)}=1$, $\tau_2^{(4)}=2$, $\tau_3^{(4)}=3$. Для кожного з них визначаємо умовно оптимальний розв'язок і відповідне до нього значення функції $F_4(\tau^{(4)})$. Для цього скористаємося рівнянням (30) і табл. 5.1, 5.2 та отриманими даними. Зокрема, для $\tau_1^{(4)}=1$ маємо:

$$F_4(\tau_1^{(4)}) = \max\begin{cases} R(\tau^{(4)}=1) & -Z(\tau^{(4)}=1)+F_5(\tau^{(5)}=2), \\ R(\tau^{(4)}=0) & -Z(\tau^{(4)}=0)-C_n+F_5(\tau^{(5)}=1), \end{cases}$$

$$= \max\begin{cases} 75 & -25+35, \\ 80 & -20-40+50, \end{cases} = 85,\ u^o = u_1.$$

$$F_4(\tau_2^{(4)}) = \max\begin{cases} 65 & -25+35, \\ 80 & -20-40+50, \end{cases} = 75,\ u^o = u_1.$$

$$F_4(\tau_3^{(4)}) = \max\begin{cases} 65 & -30+20, \\ 80 & -20-40+50, \end{cases} = 70,\ u^o = u_2.$$

Результати обчислень записуємо у вигляді табл. 5.3.

Табл. 5.3. Умовно оптимальний розв'язок для $\tau^{(4)}$

| Вік обл. $\tau^{(4)}$ | $F_5(\tau^{(4)})$ (у.о.) | $u^o$ |
|---|---|---|
| 1 | 85 | $u_1$ |
| 2 | 70 | $u_1$ |

| | | |
|---|---|---|
| 3 | 70 | $u_2$ |

Визначимо тепер умовно оптимальний розв'язок для кожного з допустимих станів обладнання на початок третього року. Вочевидь, допустимими станами є $\tau_1^{(3)} = 1$, $\tau_2^{(3)} = 2$.

$$F_3(\tau_1^{(3)}) = \max \begin{cases} R(\tau^{(3)} = 1) & -Z(\tau^{(3)} = 1) + F_4(\tau^{(4)} = 2), \\ R(\tau^{(3)} = 0) & -Z(\tau^{(3)} = 0) - C_n + F_4(\tau^{(4)} = 1), \end{cases}$$

$$= \max \begin{cases} 75 & -25 + 70, \\ 80 & -20 - 40 + 85, \end{cases} = 120, \ u^o = u_1.$$

$$F_3(\tau_2^{(3)}) = \max \begin{cases} R(\tau^{(3)} = 2) & -Z(\tau^{(3)} = 2) + F_4(\tau^{(4)} = 3), \\ R(\tau^{(3)} = 0) & -Z(\tau^{(3)} = 0) - C_n + F_4(\tau^{(4)} = 1), \end{cases}$$

$$= \max \begin{cases} 60 & -35 + 70, \\ 80 & -20 - 40 + 85, \end{cases} = 105, \ u^o = u_2.$$

Табл. 5.4. Умовно оптимальний розв'язок для $\tau^{(3)}$

| Вік обл. $\tau^{(3)}$ | $F_5(\tau^{(3)})$ (у.о.) | $u^o$ |
|---|---|---|
| 1 | 120 | $u_1$ |
| 2 | 105 | $u_2$ |

Залишилося розглянути допустимі стани обладнання на початок другого року. Вочевидь,на даний момент часу вік обладнання може дорівнювати одному року, тому треба порівняти тільки два можливі варіанти: зберегти обладнання або виконати заміну. Маємо $F_2(\tau^{(2)}) = 120$, $u^o = u_1$.

Відповідно до умови у початковий момент встановлено нове обладнання ( $\tau_1^{(1)} = 0$ ), тому проблеми вибору між збереженням і заміною обладнання не існує: обладнання слід зберегти. Отже, умовно оптимальним розв'язком є $u_1$, а значення функції

$$F_1(\tau_1^{(1)}) = R(\tau_2^{(1)} = 0) - Z(\tau_1^{(1)} = 0) + F_2(\tau_1^{(1)} = 1) = 80 - 20 + 155 = 215.$$

Отже, максимальний прибуток підприємства може дорівнювати 215 у.о. Він відповідає оптимальному плану заміни обладнання, який дістаємо на основі даних таблиць 5.1—5.4, тобто внаслідок реалізації другого етапу обчислювального процесу, що полягає у проходженні всіх розглянутих кроків від початку першого до початку п'ятого років.

Для першого кроку розв'язання безальтернативне: слід зберегти обладнання. Тож вік обладнання на початок другого року дорівнює одному року. Тоді оптимальним розв'язком для другого року є рішення про збереження обладнання. Реалізація такого рішення призводить до того, що вік обладнання на початок третього року становитиме два роки. За такого віку (3) обладнання у третьому році слід замінити. Вік обладнання на початок п'ятого року складе два роки, тобто змінювати обладнання на початок п'ятого року недоцільно.

Отже, знайдено оптимальний план заміни обладнання. У перший, другий, четвертий та п'ятий роки слід зберегти наявне обладнання, а на початку третього року -- виконати заміну обладнання.

**Приклад 2.**

У розпорядженні міністерства, якому підпорядковуються $k$ підприємств, виділені кошти у розмірі $K$ у.о. для використання їх на розвиток підприємств упродовж $m$ років. Ці кошти на початку кожного року (тобто у моменти $t_1, t_2, \ldots, t_m$ ) розподілено між підприємствами. Одночасно з цим між підприємствами розподіляється прибуток, отриманий ними за минулий рік. Таким чином, на початку кожного $i$ -го року $j$ -те підприємство отримує у своє розпорядження $x_i^{(j)}$ у.о.

Визначити такі значення $x_i^{(j)}$, тобто знайти такі розподіли коштів і прибутку між підприємствами, за яких за $m$ років забезпечується отримання максимального прибутку всіма підприємствами.

*Розв'язання*.

Нехай $j$ -му підприємству на $i$ -й рік виділяється $x_i^{(j)}$ у.о. Будемо розглядати цей розподіл коштів як реалізацію декого керування $u_i$. Керування полягає у тому, що на $i$ -му кроці першому підприємству виділяється $x_i^{(1)}$ уо., другому -- $x_i^{(2)}$ тощо. Сукупність чисел $x_i^{(j)}$, $j = \overline{1,k}$, визначає усю сукупність керувань $u_1, u_2, \ldots, u_m$ на $m$ кроках розподілення коштів як $m$ точок у $k$ -вимірному просторі.

Критерієм оцінки якості вибраного розподілу коштів, тобто керувань, що реалізуються, обрано сумарний прибуток за $m$ років, який залежить від усієї сукупності керувань: $W = W(u_1, u_2, \ldots, u_m)$. Отже, завдання полягає у виборі таких керуючих впливів $u_i^*$ (тобто у такому розподілі коштів), за яких функція $W$ набуває максимальне значення.

Бачимо, що це завдання є багатоетапне. Ця багатоетапність диктується її умовами, що передбачають прийняття рішень на початку кожного року періоду, що розглядається. Разом з тим, у низці інших завдань динамічного програмування багатоетапність безпосередньо не випливає із умов.

Наведемо приклад і дамо формулювання завдання у термінах загального завдання динамічного програмування.

Нехай $S = 7$ млн. у.о., $n = 3$, а значення $X_i$ і $f(X_i)$ наведено у табл. 5.5, $f(X_i)$ -- приріст випуску продукції залежно від обсягів капіталовкладень $X_i$.

Для розв'язання даного завдання динамічного програмування слід скласти рекурентне співвідношення Беллмана. У даному випадку це співвідношення призводить до таких функціональних рівнянь:

$$\begin{aligned} \varphi_1(X) &= \max_{0 \le X_1 \le X} \{f_1(X_1)\}; \\ \varphi_2(X) &= \max_{0 \le X_2 \le X} \{f_1(X_2) + \varphi_1(X - X_2)\}; \ldots; \\ \varphi_{n-1}(X) &= \max_{0 \le X_{n-1} \le X} \{f_{n-1}(X_{n-1}) + \varphi_{n-2}(X - X_{n-1})\}. \end{aligned} \qquad (32)$$

Тут функції $\varphi_i(X)$, $i = \overline{1, n-1}$, визначають максимальний приріст випуску продукції за відповідних розподілів $X$ у.о. капіталовкладень між $i$ підприємствами. Тому значення функції $\varphi_n(X)$ обчислюється лише для одного значення $X = S$, оскільки обсяг капіталовкладень, які виділяються для усіх $n$ підприємств, дорівнює $S$ у.о.

Табл. 5.5. Вхідні дані

| $X_i$ (у.о.) | $f_1(X_1)$ | $f_2(X_2)$ | $f_3(X_3)$ |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 100 | 30 | 50 | 40 |
| 200 | 50 | 80 | 50 |
| 300 | 90 | 90 | 110 |
| 400 | 110 | 150 | 120 |
| 500 | 170 | 190 | 180 |
| 600 | 180 | 210 | 220 |
| 700 | 210 | 220 | 240 |

Згідно із (32) і даними табл. 5.5 шукатимемо спочатку умовно оптимальні, а потім оптимальні розподіли капіталовкладень між підприємствами.

Почнемо із визначення умовно оптимальних капіталовкладень, що виділяються на розвиток першого підприємства. Для цього знаходимо значення $\varphi_1(X)$ для кожного

$$X = \{0, 100, 200, 300, 400, 500, 600, 700\}. \qquad (33)$$

Нехай $X = 0$, тоді $\varphi_1(0) = 0$. Візьмемо тепер $X = 100$. Тоді за даними табл. 5.5 дістанемо

$$\varphi_1(100) = \max\{0, 30\} = 30, \; X_1^o = 100.$$

Аналогічно знаходимо уовно оптимальні розв'язки для інших значень $X$:

$$\varphi_1(200) = \max\{0, 30, 50\} = 50, \; X_1^o = 200;$$

$$\varphi_1(300) = \max\{0, 30, 50, 90\} = 90, \; X_1^o = 300;$$

$$\varphi_1(400) = \max\{0, 30, 50, 90, 110\} = 110, \; X_1^o = 400;$$

$$\varphi_1(500) = \max\{0, 30, 50, 90, 110, 170\} = 170, \; X_1^o = 500;$$

$$\varphi_1(600) = \max\{0, 30, 50, 90, 110, 170, 180\} = 180, \; X_1^o = 600;$$

$$\varphi_1(700) = \max\{0, 30, 50, 90, 110, 170, 180, 210\} = 210, \; X_1^o = 700;$$

Результати обчисленьі відповідні умовно оптимальні розв'язки записуємо у табл. 5.6

Табл. 5.5. Умовно оптимальний розв'язок

| $X_1$ (тис.у.о.) | $\varphi_1(X)$ | $X_1^o$ |
|---|---|---|
| 0 | 0 | 0 |
| 100 | 30 | 100 |
| 200 | 50 | 200 |
| 300 | 90 | 300 |
| 400 | 110 | 400 |
| 500 | 170 | 500 |
| 600 | 180 | 600 |
| 700 | 210 | 700 |

Використаємо тепер дані табл. 5.5, 5.5. Умовно оптимальні обсяги капіталовкладень, що виділяються другому підприємству. Знайдемо

$$\varphi_2(X) = \max_{0 \le X_2 \le X} \{ f_2(X_2) + \varphi_2(X - X_2) \}$$

для кожного з допустимих значень $X$ ,b що дорівнюють (33):

$$\varphi_2(X) = 0, \quad X_2^O = 0;$$

$$\varphi_2(100) = \max\{0+30; 50+0\} = 50; \quad X_2^o = 100;$$

$$\varphi_2(200) = \max\{0+50; 50+30; 80+0\} = 80; \quad X_2^o = 100;$$

$$\varphi_2(300) = \max\{0+90; 50+50; 80+30; 90+0\} = 100; \quad X_2^o = 200;$$

$$\varphi_2(400) = \max\{0+110; 50+90; 80+50; 150+0\} = 150; \quad X_2^o = 400;$$

$$\varphi_2(500) = \max\{0+170; 50+110; 80+90; 150+30; 190+0\} = 190; \quad X_2^o = 500;$$

$$\varphi_2(600) = \max\{0+180; 50+170; 80+110; 150+50; 190+30; 210+0\} = 220; \quad X_2^o = 100;$$

$$\varphi_2(700) = \max\{0+210; 50+180; 80+170; 150+90; 190+50; 210+30; 220+0\} = 250;$$

$$X_2^o = 200.$$

Отримані умовно оптимальні обсяги капіталовкладень, що призначаються другому підприємству, записуємо у вигляді табл. 5.7. Тут $X$ -- обсяги капіталовкладень, що призначаються першому і другому підприємствам; $\varphi_2(X)$ -- максимальний приріст випуску продукції; $X_2^o$ -- умовно оптимальний обсяг капіталовкладень, що виділяється другому підприємству.

Переходимо до відшукання значень

$$\varphi_3(X) = \max_{0 \le X_3 \le X} \{ f_3(X_3) + \varphi_3(X - X_3) \}$$

із використанням даних табл. 5.5, 5.7

Табл. 5.7. Умовно оптимальний розв'язок

| $X$ (тис. у.о.) | $\varphi_2(X)$ | $X_2^o$ |
|---|---|---|
| 0 | 0 | 0 |
| 100 | 50 | 100 |
| 200 | 80 | 100 |
| 300 | 110 | 200 |
| 400 | 150 | 400 |
| 500 | 190 | 500 |
| 600 | 220 | 100 |
| 700 | 250 | 200 |

Оскільки у даному випадку число підприємств дорівнює трьом, виконуємо обчислення лише для одного значення $X = 700$.

$$\varphi_3(700) =$$

$$\max\{0+50; 40+220; 50+190; 110+150; 120+110; 180+80; 220+50; 240+0\} = 270;$$

$$X_3^o = 600.$$

Отже, максимальний приріст випуску продукції складає 270 тис. у.о. Це станеться тоді, коли 3-му підприємству призначається 600 тис. у.о., а першому і другому -- 100 тис. у.о.

## 5.2 Метод ДП для неперервних систем

### 5.2.1 Автономна система

Нехай рух об'єкту описується рівнянням

$$\frac{dx}{dt} = f(x,u), \tag{34}$$

$x = [x_1, x_2, \ldots, x_n]$ -- $n$-вимірний вектор стану; $u = [u_1, u_2, \ldots, u_m] \in U$ -- $m$-вимірний вектор управління; $f$ -- $n$-вимірний вектор. Функціонал

$$I = \int_0^T G(x,u)dt. \tag{35}$$

Вважаємо фіксованими початковий стан $x$ і кінцевий стан $x^*$. Час переходу з початкового стану у кінцевий не фіксується. Оскільки мета оптимізації полягає в отриманні оптимальної синтезуючої функції (оптимальної стратегії), початковою точкою $x$ може бути будь-яка точка фазового простору.

Мінімальне значення функціоналу однозначно визначається початковим значенням $x$. Позначимо мінімальне значення функціоналу $S(x) = S(x_1, X_2, \ldots, x_n)$.

Нехай $x(t)$, $0 \le t \le T$ -- оптимальна траєкторія, що переводить фазову точку із початкового положення $x(0) =| bfx$ у кінцеву точку $x^*$. Тоді

$$S(x) = \min_{u(t)\in U} \int_0^T G(x(t),u(t))dt. \tag{36}$$

Подамо функціонал у вигляді

$$\int_0^T G(x(t),u(t))dt = \int_0^\Delta G(x(t),u(t))dt + \int_\Delta^T G(x(t),u(t))dt. \tag{37}$$

Нехай в інтервалі $(0,\Delta)$ вибрано управління $u(t)$, а потім відповідно до принципу оптимальності вибирається оптимальне управління. Тоді

$$S(x(\Delta)) = \int_\Delta^T G(x(t),u(t))dt. \tag{38}$$

У силу неперервності траєкторії $x(t)$ $x(\Delta) = x(0) + x(0)\cdot\Delta + j(\Delta)$.

Із урахуванням рівняння (34) маємо

$$x(\Delta) = x + f(x,u)|_{t=0} \cdot\Delta + o(\Delta).$$

Отже,

$$\int_\Delta^T G(x(t),u(t))dt = S(x + f(x,u\cdot\Delta + o(\Delta)).$$

$$\int_0^\Delta G(x(t),u(t))dt = G(x,u)\cdot\Delta + o(\Delta).$$

Функціонал (37) набуває значення

$$S(x) = \min_{u(t)\in U}[G(x,u)\cdot\Delta + o(\Delta) + S(x + f(x,u\cdot\Delta + o(\Delta))]. \tag{39}$$

Оскільки функція $S(x)$ припускається неперервно диференційовною,

$$S(x+f(x,u\cdot\Delta+o(\Delta))=S(x)+\frac{dS}{dx}f(x,u)\cdot\Delta+o(\Delta).$$

За правилами диференціювання скалярної функції по векторному аргументу

$$\frac{dS}{dx}=\left(\frac{\partial S}{\partial x_1},\frac{\partial S}{\partial x_2},\ldots,\frac{\partial S}{\partial x_n}\right).$$

Із (**Ошибка! Источник ссылки не найден.**) отримуємо

$$0=\min_{u\in U}\left[G(x,u)+\frac{dS}{dx}f(x,u)\right]\Delta+o(\Delta). \tag{40}$$

Здійснимо граничний перехід за $\Delta\to 0$:

$$0=\min_{u\in U}\left[G(x,u)+\frac{dS}{dx}f(x,u)\right]. \tag{41}$$

Це -- *функціональне рівняння Беллмана.* До нього треба приєднати межову умову

$$S(x^*)=0. \tag{42}$$

Розв'язання задачі (41), (42) разом із функцією $S(x)$, яка задає залежно від початкової точки $x$ мінімальне значення функціоналу, визначається також функція $u(x)$, що задає оптимальну траєкторію, або оптимальну синтезуючу функцію.

*Рівняння Беллмана задає необхідну умову мінімуму.*

*Приклад.* Розглянемо об'єкт, рух якого задається рівнянням

$$\frac{dx_1}{dt}=x_2,\quad \frac{dx_2}{dt}=u.$$

В якості кінцевої точки $x^*$ виберемо початок координат, тобто $x^*=0$. Якість процесу управління будемо оцінювати функціоналом

$$I=\int_0^T[x_1^2+x_2^2+u^2]dt. \tag{43}$$

Отже, мова йде про визначення оптимальної стратегії $u=f(x_1,x_2)$, яка забезпечить переведення фазової точки із довільного початкового стану у початок координат, й при тому так, щоб на траєкторії руху функціонал (43) набував найменше значення.

Запишемо функціональне рівняння Беллмана

$$0=\min_u\left[x_1^2+x_2^2+u^2+\frac{\partial S}{\partial x_1}x_2+\frac{\partial S}{\partial x_2}u\right]. \tag{44}$$

Оскільки на параметр u відсутні будь-які обмеження, для визначення мінімуму необхідно продиференціювати рівняння (44) по $u$:

$$2u+\frac{\partial S}{\partial x_2}\to u=-\frac{1}{2}\frac{\partial S}{\partial x_2}. \tag{45}$$

Рівняння Беллмана набуває вигляду

$$x_1^2+x_2^2+\frac{\partial S}{\partial x_2}-\frac{1}{4}\left(\frac{\partial S}{\partial x_2}\right)^2=0.$$

Будемо шукати розв'язок у вигляді квадратичної форми

$$S(x)=C_1x_1^2+C_2x_1x_2+C_3x_2^2.$$

Тоді

$$\frac{\partial S}{\partial x_1} = 2C_1x_1 + C_2x_2; \quad \frac{\partial S}{\partial x_2} = C_2x_1 + 2C_3x_2.$$

Маємо

$$x_1^2 + x_2^2 - \frac{1}{4}(4C_3^2x_2^2 + 4C_2x_1x_2 + C_3^2x_1^2) + 2C_1x_1x_2 + C_2x_2^2 = 0.$$

Коефіцієнти $C_1, C_2, C_3$ визначаються із системи рівнянь

$$\begin{cases} 1 - \frac{1}{4}C_2^2 = 0, \\ 1 - C_3^2 + C_3 = 0, \\ 2C_1 - C_2C_3 = 0. \end{cases}$$

Ця система має два розв'язки:

$$C_2 = 2, C_3 = \sqrt{3}, \ C_1 = \sqrt{3};$$

$$C_2 = 2, C_3 = -\sqrt{3}, \ C_1 = -\sqrt{3}.$$

Згідно з (45) маємо дві синтезуючі функції:

$$u = -x_1 - \sqrt{3}x_2, \quad u = -x_1 + \sqrt{3}x_2.$$

Отримуємо дві лінійні системи, причому лінійна система, що породжена функцією $u = -x_1 + \sqrt{3}x_2$, виявляється нестійкою й отже не може забезпечити перехід фазової точки у початок координат.

### 5.2.2 Неавтономна система

Розглянемо неавтономну систему рівнянь

$$\frac{dx_i}{dt} = f_i(x_1, x_2, \ldots, x_n; u_1, u_2, \ldots, u_m, t), \ i = \overline{1,n}.$$

Функціонал

$$I = \int_{t_0}^{T} G(x_1, x_2, \ldots, x_n; u_1, u_2, \ldots, u_m, t)dt.$$

Розглядається задача про відшукання управління, що задане у вигляді оптимальної стратегії, яку здійснює переведення фазової точки $x = (x_1, x_2, \ldots, x_n)$ системи із заданого, але будь-якого початкового стану у деяку задану точку $x^* = (x_1^*, x_2^*, \ldots, x_n^*)$ так, щоб функціонал вище набував мінімальне значення. Початковий момент часу $t_0$ вважається фіксованим, але може бути будь-якою величиною. Кінцевий момент часу $T$ вважатимемо не заданим, а визначається із умови проходження траєкторії $x(t)$, $t_0 \le t \le T$, через точку $x^*$, тобто $x(T) = x^*$. Скористаємося способом приведення неавтономної задачі оптимального управління до автономної. Запишемо вхідне рівняння у векторній формі

$$\frac{dx}{dt} = f(x, u, t),$$

Приєднаємо до початкових рівнянь рівняння

$$\frac{dx_{n+1}}{dt} = 1, \rightarrow x_{n+1} = t;\ \ x_{n+1}(t_0) = t_0.$$

Маємо, таким чином, автономну систему диференційних рівнянь

$$\frac{dx}{dt} = f(x,u,t), \quad \frac{dx_{n+1}}{dt} = 1.$$

і функціонал вигляду

$$I = \int_{t_0}^{T} G(x,u,x_{n+1}dt.$$

Задача оптимального управління вище є задача із закріпленим лівим і рухомим правим кінцями траєкторії. Функціональне рівняння Беллмана має вигляд

$$0 = \min_{u \in U}\left[ G(x,u,x_{n+1}) + \frac{\partial S}{\partial x} f(x,u,x_{n+1}) + \frac{\partial S}{\partial x_{n+1}} \right].$$

Оскільки $x_{n+1} = t$ , рівняння вище можна записати у вигляді

$$-\frac{\partial S}{\partial t} = \min_{u \in U}\left[ G(x,u,x_{n+1}) + \frac{\partial S}{\partial x} f(x,u,x_{n+1}) \right].$$

Межова умова

$$S(x^*,T) = 0.$$

## 5.3 Контрольні запитання

1. Що є предметом динамічного програмування?
2. Сформулюйте постановку задачі динамічного програмування у загальному вигляді.
3. Сформулюйте принцип оптимальності Беллмана.
4. Чому задачі динамічного програмування називають багатокроковими?
5. З яких етапів складається алгоритм розв’язання задачі динамічного програмування?
5. Яким чином під час розв’язання задачі динамічного програмування проводиться умовна оптимізація?
7. Яким чином під час розв’язання задачі динамічного програмування проводиться безумовна оптимізація?
7. Перелічіть класи задач, розв’язувані за допомогою динамічного програмування.
8. Сформулюйте постановку задачі розподілу ресурсів методом динамічного програмування.
9. Охарактеризуйте основні етапи алгоритму розв’язання задачі розподілу ресурсів методом динамічного програмування.
10. У чому полягають переваги динамічного програмування?

# Розділ 6. Застосування методів оптимізації в біології та медицині

## 6.1. Передумови для застосування методів оптимізації

Сучасне моделювання (в тому числі аналітичне) прийнято відслідковувати від групи вчених, пов'язаних з Ньютоном, який створив теорію нескінченно малих - Теорію Диференціального і Інтегрального обчислення. Це дивовижне прозріння практично вперше породило аналітичні математичні моделі повністю відповідні реальним фізичним процесам.

Ньютон висунув геніальну по простоті і адекватності ідею, що значна кількість фізичних моделей макросвіту пов'язують у своїх взаєминах лінійні переміщення з їх швидкостями, прискореннями тощо і створив адекватний апарат опису об'єктів з даними змінними. За допомогою цієї теорії було вирішено грандіозне завдання -- завдання моделювання руху тіл небесної механіки. Паралельно ця задача породила до життя супутні напрямки досліджень - теорію ймовірності (роботи Лагранжа, Муавра, Байєса). Звичайно теорія ймовірності має свої власні корені, пов'язані з вивченням випадковості, але обчислювальні, практичні аспекти теорії ймовірності стимулювало споріднене «небесній механіці» завдання з теорії вимірювань. Саме нижче зазначена задача стала першим сформульованим завданням статистичного моделювання.

Необхідно було зуміти найточніше визначати початкові умови для вирішення завдань небесної механіки, тобто визначити найточніше розташування зірок на небосхилі за неодноразовими його вимірами.

Отже, в 1800-тих роках Лаплас, Гаус і Лежандр, кожен з яких у ті часи працював над теорією руху небесних тіл і не міг оминути проблеми теорії вимірювань, запропонували свої варіанти вирішення такого завдання. Гаусом і Лежандром було запропоновано оцінювати невідоме значення вимірюваної величини $a$ за його повторними вимірами $x_1, x_2, \ldots, x_n$, як таку величину $\hat{a} = a(x_1, x_2, \ldots, x_n)$, яка забезпечує мінімум функціоналу

$$J = \min_{\hat{a}} \sum_{i=1}^{n} (x_i - \hat{a})^2. \qquad (257)$$

Це була перша постановка для розрахунку параметра методом найменших квадратів. У той же час Лапласом був запропонований для тих самих умов задачі інший функціонал

$$J = \min_{\hat{a}} \sum_{i=1}^{n} | x_i - \hat{a} |. \qquad (258)$$

Виявилося, що таке значення $\hat{a}$ відповідає знаходженню вибіркової емпіричної медіани, тобто такому числу $\hat{a}$

$$\hat{a} = \begin{cases} \frac{1}{2}(x_k + x_{k=1}), & n = 2k, \\ x_{k+1}, & n = 2k+1, \end{cases} \qquad (259)$$

праворуч і ліворуч від якого розташована однакова кількість вимірювань.

Пізніше було показано, що завдання мінімізації суми модулів відхилень може бути вирішено за допомогою задачі лінійного програмування (ЛП-задачі). Покажемо, як можна

сконструювати таку задачу. Спершу розглянемо спільне та відмінності, що демонструють 2 відомі оптимізаційні підходи - МНК та ЛП-задачі.

### 6.1.1. Спільне та відмінності в задачах ЛП та МНК

Нагадаємо матричний запис канонічної ЛП-задачі

$$\min_x c^T x. \tag{260}$$

за обмежень

$$Ax = b,\ x \geq 0 \tag{261}$$

або у покомпонентному вигляді

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & & & \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}, \tag{262}$$

тобто маємо $m$ обмежень та $n$ змінних $x$ ($m < n$). Констатуємо спільні риси МНК та ЛП:

1. І МНК і ЛП -- різновиди задач оптимізації:

2. І МНК і ЛП застосовуються для вирішення завдань моделювання, але у разі ЛП ми можемо сформулювати більше різноманітних задач моделювання, оскільки тут застосовується природний елемент задач ЛП - обмеження на область моделювання.

Відмінні риси МНК та ЛП:

Оптимізаційна постановка МНК визначає функціонал мети як

$$\min \sum_{i=1}^{n} e_i^2 \tag{263}$$

де $e_i$ - складові вектора нев'язок МНК

$$e = Xa - y$$

умовної системи рівнянь

$$Xa = y \tag{264}$$

Ця система визначає перевизначену систему рівнянь, оскільки в МНК застосовується умова, що число обмежень-вимог $m$ (точок $X$) до рішення більше числа $n$ -- кількості змінних $a_n$ системи (тобто розмірності вектора параметрів $a$). отже, ми маємо у постановці МНК таку специфічну форму задачі оптимізації, коли у нас немає допустимих рішень, і ми знаходимо найкраще рішення з недопустимих.

А коли в задачі ЛП $m>n$, рішення не існує, оскільки область допустимих рішень є порожня. Тобто, рішення задачі оптимізації у формі задачі ЛП може бути застосовано тільки при недовизначеній системі типу (261), (262), тобто коли $m<n$ . Тоді для ЛП-задачі існує область допустимих рішень.

## 6.2. Підходи до вирішення несумісних оптимізаційних задач

Повернемося до розгляду задачі (260), (261) і нехай при цьому маємо ситуацію $m>n$ . Спроба вирішити її через задачу ЛП стикається із несумісністю (261) і відсутністю ОДР, оскільки $m > n$ . Якщо запровадити вектор нев'язки $y = Ax - b$ з (261) можна пропонувати два шляхи вирішення проблеми:

1. Варіант типу МНК:

### 6.2.1. Метод безумовної оптимізації за несумісної системи обмежень

Сформуємо критерій

$$L = v^T x + a^T y^2 \tag{265}$$

і будемо вимагати його мінімізації

$$\min_x c^T x + a^T (Ax - b)^2 \tag{266}$$

Далі подібно до МНК беремо похідні функції $L$ і, прирівнюючи їх до нуля, отримуємо систему лінійних рівнянь

$$\frac{\partial L}{\partial x_j} = 0, \quad j = \overline{1, n}. \tag{267}$$

Залишається питання балансування вимог до нашої задачі тобто складових критерію: первинного фукціоналу $c^T x$ та нев'язки $Ax - b$ . Це - питання вибору вектору параметрів *a*.

### 6.2.2. Застосування задач ЛП при несумісній системі обмежень

Знову повернемося до розгляду задачі. У випадку $m > n$ область допустимих розв'язків (ОДР) у задачі ЛП не існує, оскільки (262), не можуть бути задовільнені. Але, якщо розглянути мінімізацію вхідного функціоналу $c^T x$ і модулю вектору нев'язок $y=|Ax-b|$ можна отримати задачу вже у класі задач ЛП, що мають розв'язок. Тут маємо на увазі наступне: розглянемо задачу

$$\min_x [c^T x + \alpha^T y] \tag{268}$$

за обмежень

$$Ax - b - y \le 0, \tag{269}$$

$$Ax - b + y \le 0, \tag{270}$$

$$x \ge 0, y \ge 0. \tag{271}$$

Задача бути приведена до канонічної форми шляхом введення додаткових змінних $z_1$ , $z_2$ :

$$\begin{aligned} &\min_x [c^T x &&+\alpha^T y], \\ &Ax - b - y + z_1 &&= 0, \\ &Ax - b - y - z_2 &&= 0, \\ &x \ge 0, y \ge 0 &&, z_1 \ge 0, z_2 \ge 0. \end{aligned} \tag{272}$$

де вже кількість $m<n$ тобто у (272), а отже у (271) ОДР формально існує. Тепер повернемося до (268)-(271). Нехай на деякому кроці алгоритму ЛП-задачі одна із складових векторних нерівностей (269)-(270) набуде значення $Ax-b\geq 0$ . Тоді, щоб виконувалося обмеження типу , (269) відповідний $y$ має бути деяким додатним і таким, щоб при цьому $y\geq Ax-b$ , тільки у цьому випадку буде виконуватися нерівність (269), при цьому відповідна нерівність (270) виконується автоматично. Зауважимо, що мінімізація значення "$y$" забезпечує максимально можливе виконання системи рівностей (261), бо відповідає мінімізації відповідних нев'язок.

Нехай тепер на деякому кроці алгоритму у одній із складових нерівностей (269)-(270) $Ax - b$ приймає значення менше 0, тоді для виконання (270) відповідне значення "y" у нерівності типу (270) має також бути додатнім і таким що перевищує ($Ax$ - $b$), тільки тоді нерівність (270) виконається і при цьому зміст функціоналу задачі забезпечує мінімізацію "y". При цьому нерівності типу (269) виконуються автоматично. Отже, підводячі підсумок: задача (268-271) забезпечує мінімізацію вхідного функціонаалу $c^Tx$ та модулю вектора нев'язок $y=|Ax - b|$ задачі (260)-(261)

### 6.2.3. Завдання моделювання з мінімізацією критерію суми модулів нев'язок

Сформулюємо завдання апроксимації даних, виходячи з критерію мінімуму суми модулів нев'язок виходу та моделі. Нехай маємо вхідні дані

$$\begin{bmatrix} y_1 & x_{11} & x_{12} & \cdots & x_{1m} \\ y_2 & x_{21} & x_{22} & \cdots & x_{2m} \\ \cdots & & & & \\ y_n & x_{n1} & x_{n2} & \cdots & x_{nm} \end{bmatrix}$$

для відшукання параметрів $r_0, r_1, \ldots, r_m$ моделі

$$y = r_0 + \sum_{i=1}^{m} r_i x_i$$

за критерієм мінімуму суми модулів нев'язок моделі. Формально таку задачу можемо записати як:

$$L = \min_{r,s} \sum_{j=1}^{n} \left| s_j \right|$$

$$y_j - (r_0 + \sum_{i=1}^{m} r_i \cdot x_{ij}) = s_j, \qquad j = 1,...,n$$

Вочевидь, дану задачу можно записати як окремий випадок попереднього завдання (268)-(271):

$$L = \min_{r,s} \sum_{j=1}^{n} s_j$$

$$y_j - (r_0 + \sum_{i=1}^{m} r_i \cdot x_{ij}) \le s_j,$$

$$y_j - (r_0 + \sum_{i=1}^{m} r_i \cdot x_{ij}) \ge -s_j,\ j = 1,...,n$$

$$s_j \ge 0$$

Дану конструкцію подвійних нерівностей доцільно застосовувати і у задачах оптимізації, де критерій формується не як мінімум, чи максимум критеріальної змінної, а як мінімум її відхилення від заданого значення.

## 6.2.4. Завдання моделювання з одностороннім критерієм та критерієм мінімаксу

Тепер можемо легко записати цілу серію нових конструкцій для різних доцільних задач моделювання:

Задача моделювання з умови мінімума максимального відхилення моделі від табличних точок:

$$\begin{aligned} L = &\ \min_{r,x} s, \\ y_j &- \left( r_0 + \sum_{i=1}^{m} r_i x_{ij} \right) \le s, \\ y_j &- \left( r_0 + \sum_{i=1}^{m} r_i x_{ij} \right) \ge -s, \quad j = \overline{1,n} \\ s &\ge 0. \end{aligned}$$

Задача для односторонньої мінімізації:

$$\begin{aligned} L = &\ \min_{r,x} \sum_{j=1}^{n} s_j, \\ y_j &- \left( r_0 + \sum_{i=1}^{m} r_i x_{ij} \right) \le 0, \\ y_j &- \left( r_0 + \sum_{i=1}^{m} r_i x_{ij} \right) \ge -s_j, \quad j = \overline{1,n} \\ s_j &\ge 0. \end{aligned}$$

Задача для одностороннього мінімаксу:

$$
\begin{aligned}
L = \quad & \min_{r,x} \sum_{j=1}^{n} s_j, \\
y_j \quad & -\left( r_0 + \sum_{i=1}^{m} r_i x_{ij} \right) \le s_j, \\
y_j \quad & -\left( r_0 + \sum_{i=1}^{m} r_i x_{ij} \right) \ge 0, \quad j = \overline{1,n} \\
s_j \quad & \ge 0.
\end{aligned}
$$

Відмітимо, що вище не застосовані обмеження на додатність параметрів моделі, оскільки оминути це обмеження завжди можливо очевидною заміною $r_i = u_i - v_i$, де $u_i \ge 0$, $v_i \ge 0$. Наведені вище формалізації ЛП-завдань для односторонніх та мінімаксних критеріїв очевидним чином можуть бути ефективно застосовані в задачах сегментації зображень при виділенні уражених областей органів та систем організму біологічних об'єктів, задачах апроксимації меж областей. Пропонується навести приклади таких задач самостійно.

## 6.3. Розрахунок оптимальної виробничої програми фармацевтичного підприємства з екологічними обмеженнями

Розглянемо денний цикл виробництва фармацевтичного підприємства, яке може виробляти $n$ видів лікарських препаратів, причому для виробництва кожного $i$-го виду препарату необхідно використовувати пару видів сировини у співвідношенні їх ваг $-k_i$. Надалі покажемо, що постановка легко узагальнюється на довільну кількість складових для кожного виду препарату. Припускаємо, що виробничі потужності досить великі, щоб виробництво було обмежене лише екологічними обмеженнями.

Позначимо: $z$ - кількість вироблених одиниць препарату $i$-го типу; $x_i, y_i$ - витрати видів пар сировини, з яких виробляється виріб вагою $w_i$. Тут для узагальнення досить запровадити крім $x_i, y_i$ додаткові змінні, наприклад $t_i f_i$. $w_i$ - доцільно обрана вагова одиниця розфасовки при реалізації препарату, наприклад, 1 кг або 10 літрів, тощо; $u_i$ - витрати (крім витрат на закупівлю сировини) на виробництво одиниці препарату $i$-го типу; $c_i, d_i$ - ціна одиниці ваги сировини $x_i, y_i$, відповідно; $q_i$ - ціна одиниці виробу $i$-го виду;

$a_i, b_i$ -- коефіцієнти відходів відповідно для $x_i$ і $y_i$; $g_i$, $h_i$ - коефіцієнти попадання в середовище відходів сировини виду $x_i$ і $y_i$; $v_i$, $e_i$ - коефіцієнти утворення токсичних концентрацій на еталонному майданчику відбору проб для тих, хто потрапив у відходи сировини типу $x_i$ і $y_i$; $r_i$, $s_i$ - змінні, що характеризують перевищення граничної концентрації;

$R_{gr_i}$, $S_{gr_i}$ - граничні концентрації (ГДК), для концентрацій токсичних речовин, що виникають на майданчику підприємства для відбору проб, сформовані за рахунок вилучення у

середовище відходів сировини $x_i$ і $y_i$;

$R_{h_i}$, $S_{h_i}$ -- штрафи за одиницю перевищення граничних норм концентрацій $R_{gr_i}$, $S_{gr_i}$, які виплатить підприємство у даному циклі виробництва у разі виникнення такого перевищення;

SOK - сума оборотних коштів підприємства, які можливо вкласти у придбання сировини + витрати на виробництво + резерв оборотних коштів (змінна REZ);

ox - поточний запас оборотних коштів, ox - це не константа, а змінна: залишок оборотних засобів, які вдалося заощадити при купівлі сировини.

У даній постановці задачі:

нехтуємо ефектом синергізму (синергізм - ефект спільної токсичної дії речовин);

припускаємо, що немає проблем із реалізацією виробленої продукції, тобто кошти надходять по заявленим цінам відразу після виробництва;

метою завдання розрахунку програми виробництва є максимізація коштів на рахунку.

Тоді можна запропонувати наступний варіант оптимізації денної програми виробництва, тобто, величин: $x_i$ і $y_i$ - витрат сировини, $z_i$ - кількість одиниць готової продукції, $r_i$ і $s_i$ - значення перевищення ГДК , ox - залишок коштів на рахунку та $L$ - значення функціоналу. Ці величини є вирішенням такого завдання лінійного програмування:

$$L = \max\left\{\sum_{i=1}^{n}(q_i z_i - R_{h_i} r_i - S_{h_i} s_i) + ox\right\}.$$

$$\begin{aligned}
(1-a_i)x_i + (1-b_i)y_i &= w_i z_i,\\
x_i &= k_i y_i,\\
b_i h_i e_i y_i &= s_i \le S_{gr_i},\\
a_i g_i v_i x_i - r_I &\le R_{gr_i}, i = \overline{1,n}, \quad .\\
\sum_{i=1}^{n}(c_i x_i + d_i y_i + & u_i z_i + ox \le SOK,\\
ox &\ge \gamma REZ.
\end{aligned}$$

Тут $L$ - Оперативні кошти підприємства: денний дохід (без урахування податкового навантаження) плюс ox -- залишок оборотних коштів на рахунку. Значення $\gamma$ визначає значимість суми залишку оборотних засобів ox для підприємства. Дійсно, враховуючи, що готова продукція може виробляється тільки за рахунок використання оборотних коштів підприємства (для купівлі сировини та забезпечення інших витрат на виробництво), то дохід формується у функціоналі за рахунок зниження значення змінноїox (дивися останнє обмеження ЛП-завдання). Існування даного члена і варіація величиною необхідні через існування тимчасового розриву між витратою грошей на сировину, та отриманням доходу на рахунок. Це деяка сума -- резерв оборотних коштів, який завжди потрібний підприємству, щоб не допустити оголення рахунку.

Перші два обмеження задають виробничу програму, наступні два враховують існування екологічних обмежень та формують штрафні доданки (у функціоналі -- це його частини зі знаком мінус) при їх перевищенні, за кожну одиницю такого перевищення. Останнє обмеження формує залишок оборотних засобів на рахунку.

Отже, функціонал максимізує дохід, і його вид дозволяє врахувати:

штрафи за порушення екологічних обмежень;

необхідність заощадження оборотних коштів.

Сформулюємо питання до аналізу задачі:

1. Які змістовні зауваження можна зробити до цієї постановки? Наприклад, наскільки є адекватним механізм врахування штрафів, проблема емпіричності вибору $\gamma$, досить складне завдання з визначення (моделювання) значень параметрів $v_s$, $e_i$. Стосовно останнього зауваження, тут необхідна модель їх визначення (прогнозу) за прогнозними значеннями метеоумов -- за факторами погоди: температури $T$, тиску $D$, вологості $V$, сили вітру $F$ та інтенсивності опадів $O$.

2. Які зауваження можна зробити з математичної частини постановки? Наприклад, наближеність рішення пов'язана зі значенням $z_i$. Дійсно, оскільки це не неперервна величина, а по суті -- ціла змінна, при величинах $z_i \geq 50$, $n \geq 2$, ці похибки будуть незначні (менше 1%). При невиконанні даних умов, необхідно формулювати змішану цілочислову задачу.

3. В реальності необхідно врахувати ступінчастий характер нарахування штрафу. Дійсно, екологічні обмеження підприємства можуть формуватися жорсткішим чином, виключно за фактом потрапляння гранично допустимої кількості контрольованого відходу у довкілля. Тобто підприємство платить жорсткий значний штраф за кожний факт такого перевищення (незалежно, на скільки перевищили). Тобто завдання і тут переходить в клас змішано-цілочислової постановки.

Врахуємо далі останні два зауваження, тоді змінні $s_i$ і $r_i$ оголосимо бінарними (і введемо при них коефіцієнти, достатні для того, щоб відкинути рішення достатньо суттєво подалі від межі) і задача матиме вигляд

$$L = \max\left\{\sum_{i=1}^{n}(a_i z_i - R_{h_i} r_i - S_{h_i} s_i) + ox\right\}.$$

$$\begin{aligned}
&(1-a_i)x_i + (1-b_i)y_i = && w_i z_i,\\
&x_i = && k_i y_i,\\
&b_i h_i e_i y_i - s_i A_i && \leq S_{gr_i},\\
&a_i g_i v_i x_i - r_i D_i \leq && R_{gr_i}, i = \overline{1,n}, \quad .\\
&\sum_{i=1}^{n}(c_i x_i + d_i y_i + && u_i z_i + ox \leq SOK,\\
&ox \geq && \gamma REZ.
\end{aligned}$$

тут $z_i$ -- цілочислові; $s_i$, $r_i$ -- бінарні.

Конкретні величини $A_i$, $B_i$ не істотні, аби повертали рішення в допустиму область, оскільки величина перевищення ГДК не має великого значення -- штраф зазвичай один і той самий.

Щодо першого зауваження. Враховуючи, що величина коливань характеристик Уі та еві, що визначають інтенсивності утворення токсичних концентрацій залежно від метеоумов (температури, тиску, вологості, сили та стійкості напряму вітру, інтенсивності опадів, тощо) надзвичайно значна і може досягати 50%, сенс у розрахунку оптимальної програми є за наявності досить точного метеопрогнозу.

Аналіз такої оптимізаційної моделі в імітаційному експерименті може бути проведений

при отримання прогнозної моделі типу

$$y_{t+T} = f(y_t, y_{t-1}, \ldots, y_{t-T_y}, \overline{x}_t, \overline{x}_{t-1}, \ldots, \overline{x}_{t-T_x}),$$

де змінна $y$ -- утворення токсичних концентрацій у повітрі, $T$ -- інтервал упередження, $T_y$, $T_x$ -- інтервали врахування передісторії, склад вектора $x$: $x_1$ -- величина попадання відходів у середовище, погодні умови: $x_2$ -- вологість, $x_3$ -- температура, $x_4$ -- швидкість вітру, $x_5$ -- тиск, $x_6$ -- інтенсивність опадів. Прогнозні моделі налаштовуються при синтезі на фіксований інтервал випередження, та використовується на цьому ж інтервалі.

## 6.4. Розрахунок оптимальних індивідуальних лікувальних стратегій

### 6.4.1. Обгрунтування завдання

Одним з актуальних завдань медицини є розробка обчислювальних методик визначення стратегії лікування конкретних пацієнтів відповідно до їх історій хвороби, параметрів та характеристик поточного стану. Рішення, яке дозволяє нам перейти від загального опису управління станом біологічного об'єкту до їх конкретних реалізацій і, отже, отримати більш тонке налаштування моделі конкретного об'єкту, є актуальним підходом для багатьох застосувань, включаючи медицину. Тим не менш, в наш час невелика кількість робіт присвячена оптимізації в медицині, і спостерігається майже повна відсутність робіт, спрямованих на отримання найкращих індивідуальних рішень для процесу лікування пацієнта Підходи до розв'язання подібних задач в інших галузях (економіці, екології, безпеці) показують, що принцип визначення параметрів конкретного об'єкту на основі моделей, побудованих на певній сукупності подібних об'єктів, зазвичай використовується в задачах наближення, прогнозування, класифікації, але не в задачах розрахунку індивідуальних стратегій прийняття рішень. Це робить доцільним розглянути та запропонувати технології вирішення задач оптимізації індивідуальних стратегій лікування пацієнта.

### 6.4.2. Постановка завдання

Розглянемо об'єкт $Z$, для якого неможливо провести повторні випробування. Для опису переходу $Z$ із початкового $x^{in}$ у кінцевий стан $x^{out}$ передбачається існування моделей виду

$$q = G(x, u, x^{in}), \quad x^{out} = F(x, ux^{in}), \tag{273}$$

де $q$ - критерій стану об'єкту, $x$ - вектор стану об'єкту.

Існує область інтересу (ОІ) $x$ у початкових станах об'єкту. Перехід від ОІ $x^{in}$ до області допустимих значень $H(x^{out}) \le 0$, або в окремому випадку

$$x^{out}_{\min} \le x^{out} \le x^{out}_{\min}$$

має відбуватися під впливом вектора керуючих впливів $u$: $u_{\min} \le u \le u_{\max}$.

Необхідно знайти значення вектору $u$ (у загальному випадку послідовність $u_i$, яка

переводить об'єкт $Z$ зі стану $x^{in}$ у кінцевий стан $x^{out}: H(x^{out}) \le 0$ оптимально у сенсі критерію $q$.

Приклади об'єктів у цьому класі задач:

$Z$ - біологічний об'єкт (людина), ОІ $x^{in}$ - область початкових станів конкретного захворювання, $u^{out}$ - найкраща стратегія лікування.

$Z$ - конкретний економічний об'єкт, ОІ $x^{in}$ - область початкових станів для деякого кризового періоду, $u^{out}$ - оптимальна стратегія виходу з кризи.

Основна проблема завдання полягає у тому, що задача оптимізації переходу $Z$ із заданої $x^{in} \in OIx^{in}$ в область $Hx^{out}$ потребує знання певного набору реакцій об'єкту за різних стратегій керування при переміщенні об'єкту із ОІ $x^{in}$ у $Hx^{in}$, які містяться у матриці $M = |X||U|$, що використовується для отримання моделей (273). Але, як відмічено вище, об'єкт $Z$ не можна піддавати повторним іспитам, тобто примусовому поверненню $Z$ в область $OIx^{in}$ і використанню $u$ для переводу $Z$ в $Hx^{out}$. Необхідні пасивні спостереження можна отримати лише випадковим чином.

### 6.4.3. Шляхи і труднощі вирішення проблеми

Нагадаємо загальноприйнятий підхід до формалізації задачі оптимізації розрахунку стратегій. Розглянемо блочну матрицю спостережень $M = |X||U|$, отриману в результаті моніторингу об'єкта або активного експерименту. Припустимо, ми отримали оператор переходу для об'єкта і з початкового стану у кінцевий стан у формі $x^{out} = F(x,u,x^{in})$, згідно з даними спостереження. Відома функція якості для кінцевого стану об'єкта $q = G(x,u,x^{in})$ і обмеження на кінцевий стан для кожної координати (у загальному випадку $Hx^{out} \le 0$) і обмеження на вектор впливу . Тоді для цього об'єкту можна сформулювати задачу оптимізації у формі

$$\begin{aligned} &\mathrm{extr}_x q = && \mathrm{extr}_u G(x,u,x^{in}), \\ &x^{out}_{\min} \le && F(x^{in},x,u) \le x^{out}_{\max}, \\ &H(x^{out}) \le 0, && u_{min} \le u \le u_{\max}.. \end{aligned} \qquad (274)$$

Вочевидь, задача (274) може бути представлена як задача нелінійного програвання різної складності залежно від типу оператора $F$, складності обмежень і типу функціоналу $q$. У низці випадків (мала розмірність задачі, опуклість області допустимих рішень тощо) задачу можна ефективно розв'язати.

Зауважимо, що даний традиційний підхід реалізується лише тоді, коли є можливість виконувати повторні випробування або спостереження над об'єктом під час його переходу від початкових станів $x^{in}$ до деяких інших вихідних станів $x^{out}$, застосувавши певне значення вектора впливу $u$. Тобто, коли умови завдання не мають принципових застережень щодо проведення повторних випробувань. Тоді можна отримати матрицю об'єкт-властивості $M|X||U|$, сформувати необхідні моделі і побудувати задачу одноетапної оптимізації (274) для пошуку оптимального рішення. Але існує широкий клас об'єктів, які неможливо повторно випробувати, або для них неможливо спостерігати багаторазово подібні стани. Типовим прикладом є біологічні об'єкти (зокрема пацієнти), для яких неможливо провести повторні

клінічні випробування зі смертельною хворобою, вивчити різні режими лікувальної терапії, а потім індивідуально розробити оптимальну стратегію лікування для цього пацієнта.

Водночас, якщо є можливість проводити поодинокі випробування для достатньо однорідних груп об'єктів, фіксуючи параметри окремого об'єкта, необхідні для отримання моделей достатньої якості, то можна сформувати блочну матрицю спостережень $M$ за вибіркою об'єктів. Тоді при формуванні оптимізаційної задачі можна побудувати моделі, спільні для групи об'єктів, що розглядаються. Підставляючи надалі індивідуальні параметри об'єкту і початкові умови, ми отримуємо і розв'язуємо конкретний варіант оптимізаційної задачі, налаштованої саме для цього об'єкту, отримуючи тим самим індивідуальну стратегію.

Реалізуючи запропоновану послідовність дій, розглянемо окремий випадок задачі (274) - задачу одноетапної оптимізації, коли потрібно перевести деякий об'єкт з параметрами $p$ із заданого початкового стану $x^{in}$ у кінцевий стан $x^{out}$ із використанням оптимального значення вектора впливу $u$. Вектор кінцевого стану не вказано, для нього визначено діапазон допустимих значень, але має бути досягнуто найкраще значення критеріальної змінної $q$. У розглянутому формулюванні різниця між терміном «критеріальна змінна» і «критерій» полягає у тому, що критеріальна змінна може мати певне значення ще до прийняття рішення впливу - значення $q^{in}$. Значення змінної критерію після застосування впливу є $q^{out}$.

Отже, нехай в результаті моніторингу або активного експерименту отримано матрицю об'єкт-властивості $M$:

$$M = | P | x^{in} | q^{in} | x^{out} | q^{out} | U |, \qquad (275)$$

де $P = [p_{ij}]$, $i = \overline{1,n}$, $j = \overline{1,s}$ -- матриця параметрів об'єкту, де кожен рядок містить $s$ відповідних значень параметрів об'єкту, $x^{in} = [x_{ij}^{in}]$, $x^{out} = [x_{ij}^{out}]$, $i = \overline{1,n}$, $j = \overline{1,d}$ -- кількість змінних стану об'єкту, $q^{in} = [q_1^{in}, q_2^{in}, \ldots, q_n^{in}]^T$ -- значення вектору критеріальної змінної до та після застосування впливу, $U = [u_{ij}]$, $i = \overline{1,n}$, $j = \overline{1,h}$, $h$ -- кількість змінних впливу.

Нехай наявні дані надають можливість побудувати адекватні моделі вихідних станів об'єктів

$$x_i^{out} = F_i(p, q^{in}, x^{in}, u), \ i = \overline{1,n}, \qquad (276)$$

для змінної критерію

$$q^{out} = G(p, q^{in}, x^{in}, u), \qquad (277)$$

і для області існування розв'язку

$$H(p, q^{in}, x^{in}, u) \le 0. \qquad (278)$$

Тоді можна записати таку задачу оптимізації:

$$\begin{aligned} \text{extr}_u q^{out} = \quad & \text{extr}_u G(x, u, x^{in}), \\ x_{\min}^{out} \le \quad & F(x^{in}, x, u) \le x_{\max}^{out}, \\ H(p, q^{in}) \le 0, \quad & u_{min} \le u \le u_{\max}. \end{aligned} \qquad (279)$$

Після заміщення даних об'єкту $p, q^{in}, x^{in}$ в (279) отримуємо задачу оптимізації (280), яка орієнтована на розрахунок оптимального ефекту для вибраного об'єкту:

$$\begin{aligned} \min_u q^{out} = & \quad \min_u g_u(u), \\ x_{\min}^{out} \le & \quad f(u) + a'_0 \le x_{\max}^{out}, \\ f(u) \le & \quad 0, \\ u_{\min} \le & \quad u \le u_{\max}. \end{aligned} \tag{280}$$

Із урахуванням відомих труднощів із розв'язуванням задач нелінійного програмування, спробуємо знайти шляхи отримання (280) у лінійній формі. Тоді для вирішення цих завдань можна було б використовувати ефективний обчислювальний засіб лінійного програмування.

Але звернемо увагу на іншу проблему. Якщо об'єктами розгляду є пацієнти, то отримання адекватних моделей стикається, як правило, з недостатнім рівнем варіабельності змінних впливу в експериментальних даних. Проблеми виникають через те, що методологія клінічних досліджень спрямована на використання вибірки пацієнтів для підтвердження важливості терапевтичного ефекту, а не на отримання експериментальних даних, що забезпечують найкращу адекватність моделей. У результаті дослідники змушені мати справу або з вказаними даними клінічних досліджень, або з даними моніторингу процесу лікування при вирішенні задачі оптимізації для конкретного пацієнта.

Одним із можливих механізмів підвищення адекватності моделей у цих умовах є використання оптимізаційних моделей, налаштованих на конкретні умови застосування препаратів. Налаштування здійснюється за параметрами і початковими станами пацієнта. Зазначимо, що параметри і початкові стани об'єкта після підстановки в модель перетворюються на фіксовані значення параметрів моделі. Таким чином, для підвищення адекватності моделей можливий пошук їхніх структур, як структур оптимальної складності, які будуть нелінійними за зазначеними змінними, не виходячи водночас за межі лінійного класу задач оптимізації за змінними впливу.

Подальший розвиток підходу до розрахунку індивідуальних стратегій передбачає перехід від задач виду (279) ,(280) до задач оптимізації з ефективнішими обчислювальними процедурами. Нижче ми покажемо основні етапи запропонованого підходу та розглянемо приклад розрахунку оптимальної індивідуальної стратегії.

## 6.4.4. Розрахунок у класі лінійних індивідуальнох стратегій

Отже, маємо матрицю вихідних даних у вигляді (275). Сформуємо задачу одноетапної оптимізації для отримання кінцевого стану об'єкта в заданій області з найкращим значенням критеріальної змінної. Для отримання моделей стану об'єкта та критеріальної змінної, які є лінійними за змінними впливу, сформуємо узагальнені змінні для моделювання $x_j^{out}$, $j = \overline{1,d}$ і $q^{out}$ як набір змінних, нелінійних за $pq^{in}$ та лінійних за $u$. Тоді за допомогою алгоритмів моделювання можна отримати модель переходу від стану об'єктів $q^{in}$ до стану об'є ктів $q^{out}$ у вигляді

$$q^{out} = g(p, q^{in}, x^{in})u + g_0(p, q^{in}, x^{in})u + c_0. \tag{281}$$

$$x_i^{out} = f_i(p, q^{in}, x^{in})u + f_0(p, q^{in}, x^{in})u + a_0, \; i = \overline{1,d} \tag{282}$$

$$H_u(q^{in}, x^{in})u + H_0(q^{in}, x^{in})u + b_0 \le 0. \tag{283}$$

Задача оптимізації критерію

$$g(p,q^{in},x^{in})u + g_0(p,q^{in},x^{in})u$$

з обмеженнями на $x_j^{out}$, $j=\overline{1,d}$, $u_j$, $j=\overline{1,h}$ тепер має вигляд:

$$\begin{aligned} \min_u q^{out} = & \quad \min_u g_u(u), \\ x_{\min}^{out} \le & \quad f(u) + a'_0 \le x_{\max}^{out}, \\ f(u) \le & \quad 0, \\ u_{\min} \le & \quad u \le u_{\max}. \end{aligned} \tag{284}$$

Для розрахунку невідомих впливів об'єкту «Z» підставляємо значення його параметрів та історії хвороби в отримані моделі виду (281)-(283), налаштовуючи систему обмежень і критерій до зазначеного об'єкту. Тоді задача (281) набуває лінійного вигляду за змінними впливу. У спрощеному варіанті, коли допустима область задана обмеженнями на кожну координату окремо, отримуємо задачу у вигляді

$$\begin{aligned} \min_u q^{out} = & \quad \min_u g_c(u), \\ x_{\min}^{out} \le & \quad f(u) + a'_0 \le x_{\max}^{out}, \\ u_{\min} \le & \quad u \le u_{\max}. \end{aligned} \tag{285}$$

Таким чином формуються частково нелінійні моделі кінцесого стану об’єкту, підвищуючи точність прогнозу і при цьому задача оптимізації не виходить за межі класу задач лінійної оптимізації.

## 6.5. Контрольні питання

1. Сформулюйте задачу, що виникла, як передумова для застосування апарату МП для моделювання

2. Порівняйте методи оптимізації МНК та ЛП для моделювання, та сформулюйте області застосування

3. Наведіть та обгрунтуйте ЛП задачу з критерієм модулю відхилень

4. Наведіть та обгрунтуйте конструкції ЛП задач для моделювання з односторонніми критеріями та критерієм мінімаксу.

5. Обгрунтуйте застосування односторонніх та мінімаксних критеріїв в задачах моделювання в біології та медицині