

Л. С. КОРЯШКІНА
С. А. УС

ПРАКТИКУМ ЗА КУРСОМ

МЕТОДИ ОПТИМІЗАЦІЇ ТА ДОСЛІДЖЕННЯ ОПЕРАЦІЙ

ЧАСТИНА **2**
НЕЛІНІЙНЕ ПРОГРАМУВАННЯ
НАВЧАЛЬНИЙ ПОСІБНИК

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
"ДНІПРОВСЬКА ПОЛІТЕХНІКА"



Л.С. Коряшкіна, С.А. Ус

Практикум за курсом
«МЕТОДИ ОПТИМІЗАЦІЇ ТА ДОСЛІДЖЕННЯ ОПЕРАЦІЙ»
ЧАСТИНА II. НЕЛІНІЙНЕ ПРОГРАМУВАННЯ
Навчальний посібник

Дніпро
«Свідлер А.Л.»
2023

УДК 519.8
К 70

Затверджено до видання вченою радою університету як навчальний посібник для здобувачів ступеня бакалавра спеціальності 124 Системний аналіз (протокол № 9 від 21.09.2023).

Рецензенти:

О.Д. Станіна – канд. техн. наук, доц. (Дніпропетровський державний університет внутрішніх справ);

С.Є. Авраменко – магістр з системного аналізу, Machine Learning Engineer (IT COMPARUS.UA);

Д.О. Сердюк – магістр з системного аналізу, фахівець математик-програміст, (PROSTOR).

Коряшкіна Л.С.

К 70 Практикум за курсом «Методи оптимізації та дослідження операцій». Частина II. Нелінійне програмування: навч. посіб. / Л.С. Коряшкіна, С.А. Ус / Видавець «Свідлер А.Л.»2023. – 220 с.

ISBN 978-617-627-174-1

Міститься матеріал щодо проведення практичних занять з курсу «Методи оптимізації та дослідження операцій», пов'язаний з використанням нелінійного програмування. Для кожного заняття наведено тему, мету, план, опорні поняття, рекомендовану літературу (базову та допоміжну), інформаційні ресурси, варіанти завдань для самостійної та індивідуальної роботи студентів, поточні контрольні запитання / завдання.

ISBN 978-617-627-174-1

УДК 519.8

© Л.С. Коряшкіна, С.А. Ус

© НТУ «Дніпровська політехніка», 2023

Зміст

Передмова	5
Практичне заняття № 1	7
Побудова нелінійних математичних моделей оптимізаційних задач	7
Практичне заняття № 2	27
Графічний метод розв'язання нелінійного програмування	27
Практичне заняття № 3	36
Постановка задачі нелінійного програмування. Теорема Веєрштрасса. Задачі безумовної оптимізації	36
Практичне заняття № 4	55
Задачі з обмеженнями. Метод множників Лагранжа	55
Практичне заняття № 5	66
Методи одновимірної оптимізації	66
Практичне заняття № 6	77
Основна задача опуклого програмування. Критерій оптимальності. Розв'язання задачі умовної оптимізації квадратичної функції	77
Практичне заняття № 7	95
Задачі дробово-лінійного програмування	95
Практичне заняття № 8	105
Наближені методи оптимізації. Прямі методи безумовної оптимізації (методи нульового порядку): покоординатного спуску, пошук за деформованим багатогранником	105
Практичне заняття № 9	122
Методи нульового порядку: Розенброка, Хука і Дживса	122
Практичне заняття № 10	127
Градiєнтні методи безумовної оптимізації	127
Практичне заняття № 11	141
Методи другого порядку. Метод Ньютонa і його модифікації	141
Практичне заняття № 12	148
Методи перемінної метрики. Квазіньютонівські методи	148
Практичне заняття № 13	157
Задачі умовної оптимізації. Метод проєкції градієнта. Метод умовного градієнта	157
Практичне заняття № 14	167

Задачі умовної оптимізації. Виключення обмежень за допомогою заміни змінних. Методи штрафних функцій	167
Практичне заняття № 15	187
Евристичні алгоритми оптимізації	187
ДОДАТОК А	213
Огляд тестових функцій	213
ДОДАТОК Б.....	219
Додаткова література	219

ПЕРЕДМОВА

Оскільки будівля всього світу досконала і зведена премудрим Творцем, то у світі не відбувається нічого, у чому не було б видно сенсу якогось максимуму чи мінімуму.

Леонард Ейлер

Багато завдань, що виникають в фундаментальних науках (фізиці, хімії, молекулярній біології чи інших), а також в багатьох прикладних дисциплінах, зводяться до задач нелінійного програмування, тобто оптимізації, де цільова функція та/або обмеження на змінні задаються у вигляді нелінійних функцій.

Перші такі задачі геометричного змісту з'явилися ще в стародавні часи. Розвиток промисловості XVII – XVIII століть викликав необхідність дослідження більш складних завдань на екстремум, появу варіаційного числення. В XX столітті в умовах величезного розмаху виробництва і усвідомлення обмеженості ресурсів планети виникла задача оптимального використання енергії, матеріалів, робочого часу. Значну увагу стали приділяти питанням найкращого у тому чи іншому сенсі управління різними процесами фізики, техніки, економіки та ін., навіть потребам розвитку самої обчислювальної техніки. Розв'язання таких задач є складним з обчислювальної точки зору, оскільки цільова функція може мати кілька локальних та глобальних оптимумів, що прийнято характеризувати як багатоекстремальність або мультимодальність. Іншими труднощами є нелінійність, недиференційовність, яружність, висока розмірність простору пошуку, висока обчислювальна складність та інше.

Традиційно наближені методи оптимізації в багатовимірному просторі діляться на три великі групи:

1) Прямі методи (методи нульового порядку) засновані на порівнянні значень цільової функції в різних точках (покоординатний спуск, симплекс-метод). Стратегія прямих методів – поступове наближення до оптимуму, а непрямих – прагнення знайти розв'язок, не досліджуючи неоптимальні точки.

2) Градієнтні методи засновані на знанні перших похідних цільової функції. Оптимізація, в основу якої покладено ідею руху по найкрутішій стежці, є методом найшвидшого спуску. Вектор градієнта є перпендикулярним до лінії рівня і вказує напрямком до нової точки в просторі проектування.

3) Методи другого порядку застосовують також і похідні другого порядку. Досліджувати другі похідні функції доводиться задля з'ясування, чи є дана точка мінімумом, максимумом або сідлом.

З розвитком прикладних наук дослідники все частіше стикаються із зростаючою складністю об'єктів, що оптимізуються. У такій ситуації цілком природним є й ускладнення їх математичних моделей, що, як наслідок, створює перешкоди під час пошуку оптимальної комбінації параметрів. Часто її

аналітично знайти неможливо, і виникає необхідність побудови нових обчислювальних методів.

Проблема наближеного розв'язання задач оптимізації, в свою чергу, пов'язана зі значними труднощами, тому нині активно розвивається парадигма обчислень, в якій обчислювальні проблеми вирішуються на основі наслідування природних явищ. В оптимізації цей підхід породив напрямок так званих еволюційних алгоритмів (еволюційне програмування, еволюційні стратегії, генетичні алгоритми) з цілою низкою оптимізаційних алгоритмів, заснованих на різних фізичних, хімічних і біологічних явищах.

Наближені методи глобальної оптимізації істотно відрізняються від стандартних локальних методів, які не можуть дати глобальний (найкращий) розв'язок задач, приміром, через багатоекстремальність цільової функції та нездатність покинути зони притягіння локальних оптимумів.

Даний посібник містить методичні матеріали, які стосуються методів розв'язання нелінійних задач математичного програмування, не лише класичних, збіжність яких за певних умов теоретично обґрунтована, а й тих, що містять евристики.

Перелік і зміст тем, наведених у книзі, визначені навчальною робочою програмою дисципліни «Методи оптимізації і дослідження операцій», розробленою в межах освітньо-професійної програми підготовки бакалаврів із спеціальності 124 Системний аналіз. Посібник також може використовуватися при підготовці бакалаврів спеціальностей 121 – Програмна інженерія та 122 – Комп'ютерні науки.

Практичне заняття № 1.

Побудова нелінійних математичних моделей оптимізаційних задач

Мета: набуття і закріплення навичок побудови нелінійних економіко-математичних моделей

План заняття

1. Загальна постановка задачі оптимізації. Математичне програмування. Класифікація задач і методів МП.
2. Побудова математичних моделей прикладних задач.
 - 2.1. Оптимізація параметрів геометричних фігур та тіл.
 - 2.2. Оптимальне багатоетапне змішування.
 - 2.3. Задача про вибір портфеля цінних паперів.
 - 2.4. Планування роботи гідроенергетичного комплексу.
 - 2.5. Геометричні завдання, які зводяться до задач нелінійного програмування.

Опорні поняття

1. Загальна постановка задачі оптимізації. Математичне програмування. Класифікація задач і методів МП

Загальну постановку задачі оптимізації можна сформулювати у такий спосіб.

Припустимо, розглядається цільова функція $f: \Omega \rightarrow \mathbb{R}$, неперервна в обмеженій області, яка визначена у такий спосіб:

$$\Omega = \{x \in D: g_i(x) \leq 0, 1 \leq i \leq m; g_i(x) = 0, m + 1 \leq i \leq s\},$$

$$D = \{x \in \mathbb{R}^N: a_j \leq x_j \leq b_j, 1 \leq j \leq N\},$$

де N – розмірність задачі, функція $g_i(x), i \leq i \leq m$ – обмеження, задані у формі нерівностей, Ω – допустима область або допустима множина розв'язків, D – область пошуку.

Тоді задача оптимізації полягає у пошуку розв'язку x^* , який забезпечує цільовій функції найменше значення на всій допустимій області, тобто

$$\min_{x \in \Omega} f(x) = f(x^*) = f^*.$$

У випадку задачі глобальної оптимізації без обмежень допустима область Ω являє собою весь N -вимірний дійсний векторний простір \mathbb{R}^N , а задача має таку форму:

$$\min_{x \in \mathbb{R}^N} f(x) = f(x^*) = f^*.$$

Це означає, що виконується наступна умова оптимальності:

$$f(x^*) \leq f(x) \quad \forall x \in \mathbb{R}^N.$$

Умова $x \in D$ приналежності допустимої точки множині D називається *прямим обмеженням*, а обмеження-рівності та обмеження-нерівності у визначенні множини Ω – *функціональними обмеженнями*.

Іноді, формулюючи математичну постановку оптимізаційної задачі, вдається уникнути включення в модель нелінійних функцій і отримати модель лінійного програмування. Але, прагнучі до врахування якомога більшої кількості параметрів і властивостей складних об'єктів чи процесів, що розглядаються, результуючу модель частіше за все одержують з класу *нелінійного програмування*.

Зауважимо, що описана задача відноситься до *математичного програмування*.

Математичне програмування – це прикладна математична дисципліна, предметом якої є дослідження екстремальних (оптимізаційних) задач та розроблення методів їхнього розв'язування. Залежно від вигляду функцій f і g_i математичне програмування можна розглядати як сукупність кількох самостійних дисциплін, що вивчають і розробляють методи розв'язання певних класів задач. Відповідну класифікацію подано на рис. 1.1. Визначимо основні характеристики задач.

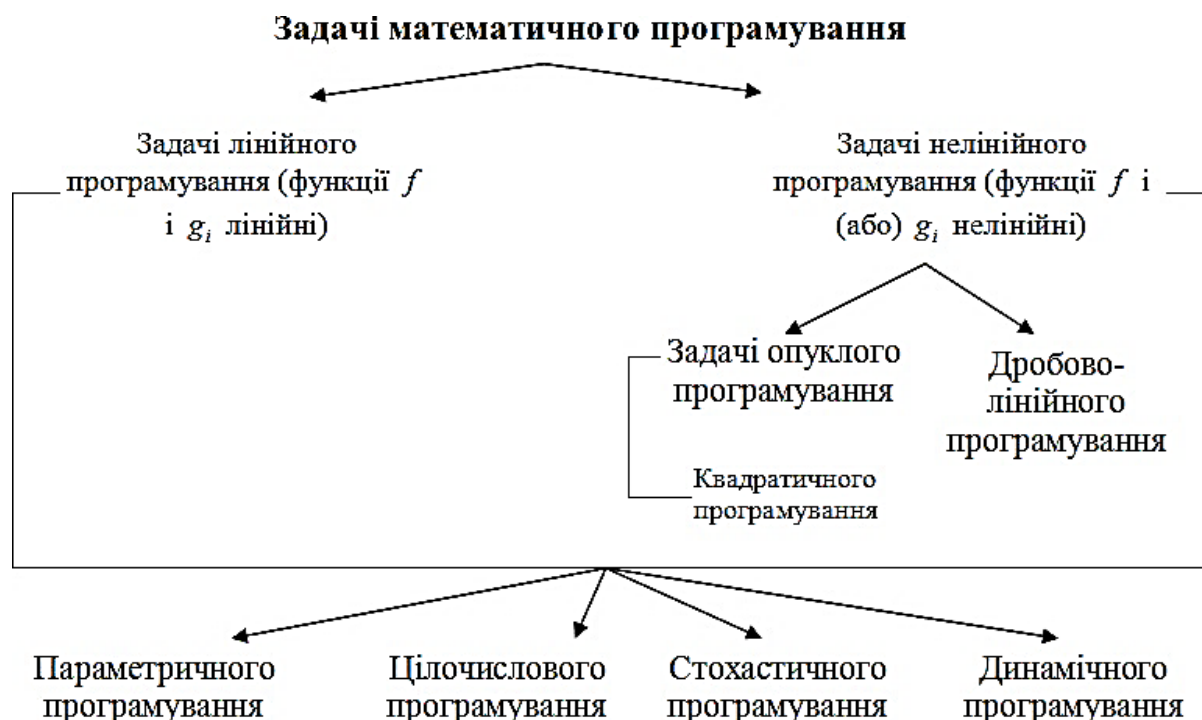


Рис. 1.1. Класифікація задач математичного програмування

Задачі опуклого програмування характеризуються тим, що їх розв'язком є максимум опуклої чи мінімум увігнутої функції.

У *квадратичному програмуванні* досліджуються задачі, де необхідно знайти максимум чи мінімум квадратичної функції за умов, що її змінні

задовольняють деяку систему лінійних обмежень (нерівностей або рівнянь).

Якщо цільова функція задачі є часткою двох лінійних функцій – отримуємо задачу *дробово-лінійного програмування*.

Параметричне програмування вивчає задачі, в яких цільова функція або коефіцієнти системи обмежень залежать від деяких параметрів.

Коли цільова функція або система обмежень містять випадкові величини, то отримана задача буде з класу *стохастичного програмування*.

І, нарешті, задачі, процес розв'язання яких є багатоетапним, досліджують у *динамічному програмуванні*.

Комбінаторний аналіз є інструментом розв'язання задач вибору і розміщення елементів деякої, як правило, скінченної, множини відповідно до заданих правил. В основу комбінаторних методів дослідження операцій покладено ідею перебору всіх допустимих розв'язків. Найбільш відомим з них є метод гілок і меж.

Статистичне моделювання досліджує процеси поведінки ймовірнісних систем за умов, коли внутрішні взаємодії у цих системах невідомі. Завдання статистичного моделювання полягає у відтворенні досліджуваного фізичного процесу за допомогою ймовірнісної математичної моделі і обчисленні характеристик цього процесу.

У дослідженні операцій методи статистичного моделювання використовуються для розв'язування задач керування запасами, аналізу систем масового обслуговування та ін.

Розглянемо приклади задач, математичні моделі яких включають умови з нелінійними залежностями або мають нелінійну цільову функцію.

2. Побудова математичних моделей прикладних задач

2.1. Оптимізація параметрів геометричних фігур та тіл

Змістова постановка задачі. Потрібно спроектувати прямокутну конструкцію у формі паралелепіпеда з відкритою передньою стінкою, об'єм якої дорівнює 16000 м^3 . При цьому периметр її основи не має перевищувати 220 м, а глибина і ширина не можуть бути більшими за 60 і 70 м відповідно. Крім того, відомо, що ширина не може перевищувати потроєну глибину, а висота – $2/3$ ширини. Вартість гофрованого матеріалу, з якого виготовляють дах та три стіни конструкції, становить 30 грн за м^2 . Потрібно визначити розміри конструкції, за яких вартість витрачених на будову матеріалів буде мінімальною.

Математична постановка задачі. Позначимо через x_1 , x_2 , x_3 глибину, ширину та висоту конструкції відповідно, що вимірюється в метрах. Тоді загальна вартість матеріалу буде включати такі складові: вартість даху – $30x_1x_2$ грн, задньої стінки – $30x_2x_3$ грн, бічних стінок – $2(30x_1x_3)$ грн. А, отже, цільова функція задачі набуває такого вигляду:

$$f(x) = 30x_1x_2 + 30x_2x_3 + 60x_1x_3.$$

Вимога стосовно об'єму конструкції визначає обмеження у вигляді такої рівності:

$$x_1 x_2 x_3 = 16000 \text{ м}^3.$$

Умови на лінійні розміри записуються у вигляді наступних нерівностей.

$$2(x_1 + x_2) \leq 220 \text{ м},$$

$$x_2 \leq 3 x_1,$$

$$x_3 \leq \frac{2}{3} x_2.$$

Крім того,

$$0 \leq x_1 \leq 60 \text{ фут},$$

$$0 \leq x_2 \leq 70 \text{ фут}.$$

Отже, у стандартній формі запису задача матиме такий вигляд:

$$f(x) \rightarrow \min$$

за умов:

$$h_1(x) = x_1 x_2 x_3 - 16000 = 0,$$

$$g_1(x) = x_1 + x_2 - 110 \leq 0,$$

$$g_2(x) = x_2 - 3 x_1 \leq 0,$$

$$g_3(x) = x_3 - \frac{2}{3} x_2 \leq 0,$$

$$0 \leq x_1 \leq 60,$$

$$0 \leq x_2 \leq 70,$$

$$0 \leq x_3.$$

Питання для обговорення

1. Якими методами можна розв'язати сформульовану математичну задачу?
2. Перевірте розмірність отриманої моделі.
3. Яким чином потрібно змінити модель, коли необхідно врахувати умову, що використовуваний гофрований матеріал закуповується листами розміром 50×40?

2.2. Оптимальне багатоетапне змішування

Задачі на суміші утворюють важливий клас прикладних оптимізаційних завдань. Вони виникають під час вибору найкращого способу змішування вихідних інгредієнтів для отримання конгломерату із заданими властивостями, які залежать від кількості компонентів, що входять до складу вихідних матеріалів. Як правило, в цих задачах відомо вартісні характеристики інгредієнтів, і шукану суміш потрібно одержати із найменшими витратами. Для

багатопродуктових задач, де необхідно отримати кілька сумішей, характерним є критерій максимізації прибутку.

Змістова постановка задачі. Для отримання кінцевого продукту при перегонці нафти та приготуванні вугільних сумішей виникає потреба змішування кількох видів сировини під час проміжних стадій процесу. Це зумовлено обмеженістю ємностей для зберігання компонентів і необхідністю сумісного транспортування вихідних товарів. На рис. 1.2 показано спрощену схему процесу змішування двох фракцій.

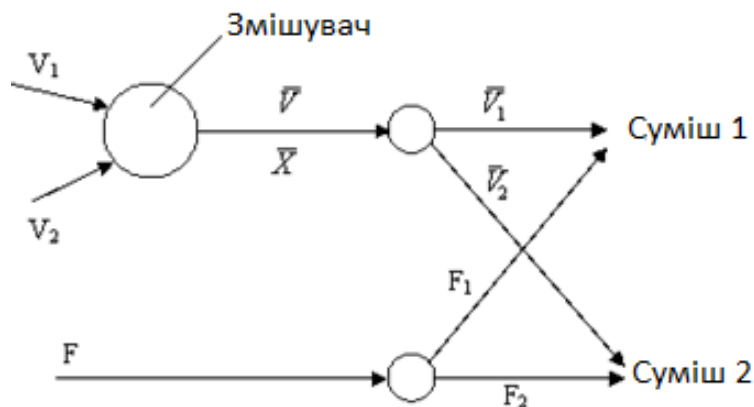


Рис. 1.2. Схема процесу змішування

В цій системі спочатку змішуються два потоки сировини: V_1 з вмістом сірки 3 % та V_2 з вмістом сірки 1 %. Отримана суміш змішується у двох різних пропорціях із третім потоком сировини F , що містить 2 % сірки. При цьому отримується 2 види кінцевої суміші. Перша суміш повинна вироблятися у кількості не більше 100 барелей за годину, причому вміст сірки не має бути більшим за 2,5 %; для другої суміші аналогічні показники мають дорівнювати 200 барелям на годину та 1,5 %. Вартість сировини для кожного з потоків 1, 2 і 3 дорівнює 6, 16 і 10 \$ за 1 барель відповідно. Ціна збуту сумішей становить 9 і 10 \$ за 1 барель відповідно. Потрібно визначити розміри потоків, які дозволять отримати максимальний дохід.

Математична постановка задачі. Припустимо, що P_i – вихід суміші i -го виду (в барелях на годину); F_i – величина потоку сировини третього виду, яка використовується в суміші i -го виду; V_i – величина потоку проміжної суміші, що використовується в i -й суміші; \bar{x}_i – вміст сірки у проміжній суміші.

Умови балансу для потоків будуть мати такий вигляд:

$$V_1 + V_2 = \bar{V}_1 + \bar{V}_2;$$

$$F_1 + \bar{V}_1 = P_1;$$

$$F_2 + \bar{V}_2 = P_2.$$

Умова балансу на вміст сірки:

$$0.03V_1 + 0.01V_2 = \bar{x}(\bar{V}_1 + \bar{V}_2);$$

$$0.02F_1 + \bar{x}\bar{V}_1 \geq 0.025P_1;$$

$$0.02F_2 + \bar{x}\bar{V}_2 \geq 0.015P_2.$$

Крім того, існує обмеження на вихід суміші, а саме:

$$P_1 \leq 100, P_2 \leq 200.$$

Вміст сірки у проміжній суміші визначається вмістом сірки у її компонентах, тому

$$0.01 \leq \bar{x} \leq 0.03.$$

Цільова функція задачі описує дохід і є лінійною, а саме:

$$f = 9P_1 + 10P_2 - 6V_1 - 16V_2 - 10F.$$

Отже, вихідна задача оптимізації має такий вигляд:

$$\text{Максимізувати } f = 9P_1 + 10P_2 - 6V_1 - 16V_2 - 10F,$$

враховуючи обмеження:

$$h_1 = V_1 + V_2 - \bar{V}_1 - \bar{V}_2 = 0;$$

$$h_2 = 0.03V_1 + 0.01V_2 - \bar{x}(\bar{V}_1 + \bar{V}_2) = 0;$$

$$h_3 = F_1 + \bar{V}_1 - P_1 = 0;$$

$$h_4 = F_2 + \bar{V}_2 - P_2 = 0.$$

$$g_1 = 0.025P_1 - 0.02F_1 - \bar{x}\bar{V}_1 \geq 0;$$

$$g_2 = 0.015P_2 - 0.02F_2 - \bar{x}\bar{V}_2 \geq 0;$$

$$P_1 \leq 100, P_2 \leq 200; 0.01 \leq \bar{x} \leq 0.03.$$

Вочевидь, нелінійними членами в цих обмеженнях є $\bar{x}\bar{V}_1$ та $\bar{x}\bar{V}_2$ в h_2 , g_1 і g_2 , решта – лінійні. Тому найпростішим підходом для розв'язання задачі є лінеаризація моделі з подальшим використанням методу лінійного програмування. З цією метою задають деяке значення \bar{x} із заданого діапазону, модель лінеаризують в околі цієї точки, і задача стає лінійною. Отриманий розв'язок задачі ЛП можна використовувати для коригування заданого значення \bar{x} , а процес розв'язання слід продовжувати до отримання задовільного наближення.

Питання для обговорення

1. Які обмеження моделі є балансовими? Обмеження якого типу (рівності, нерівності) використовуються для моделювання балансових співвідношень? Чому?

2. Розв'язати отриману задачу в пакеті MS Excel. Прокоментувати отриману відповідь.

2.3. Задачі про вибір портфеля цінних паперів

Визначення оптимального портфеля цінних паперів є однією із найважливіших задач фінансового менеджменту. Із цими задачами стикаються різні інвестиційні фірми (банки, взаємні фонди, страхові компанії). Під *портфелем цінних паперів*, як правило, розуміють розміри вкладень у різні їх види: валюта, акції та облігації, банківські депозитні сертифікати та ін.

Припустимо, що готівковий капітал C в наступному інвестиційному періоді можна вкласти у цінні папери n видів, і потрібно визначити відповідні частки вкладень.

Нехай $x_j, j = 1, 2, \dots, n$ – величина капіталу, який вкладається в цінні папери j -го виду (в доларах). Тоді змінні x_j мають такі обмеження:

$$x_1 + x_2 + \dots + x_n \leq C;$$

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

Припустимо також, що є статистичні дані щодо кожного виду вкладень за останні T років, які відображають коливання цін та виплати дивідендів протягом цього періоду. За допомогою них можна оцінити прибуток від вкладень за кожним типом цінних паперів. Позначимо через $r_j(t)$ – загальний дохід в році t на один долар вкладень в цінні папери виду j . Тоді

$$r_j(t) = \frac{|p_j(t+1) - p_j(t) + d_j(t)|}{p_j(t)},$$

де $p_j(t)$ – ціна паперів j -го типу на початок року t , а $d_j(t)$ – сумарні дивіденди, які отримано у t -му році.

Слід зауважити, що значення $r_j(t)$ не є сталими і сильно коливаються з року в рік. Вони можуть мати будь-який знак і навіть дорівнювати нулю. Тому для обґрунтування доцільності вкладень варто обчислювати *середній або очікуваний дохід* від цінних паперів виду j на один вкладений долар, а саме:

$$\mu_j = \frac{1}{T} \sum_{t=1}^T r_j(t).$$

Загальна величина очікуваного прибутку, тобто його статистичної оцінки, описується таким чином:

$$E = \sum_{j=1}^n \mu_j x_j = \mu^T x.$$

Опишемо можливі математичні моделі.

Модель I. Включає максимізацію загального очікуваного доходу (оцінка математичного сподівання доходу) за обмежень на загальний обсяг інвестицій і має такий вигляд:

$$Z = \sum_{j=1}^n \mu_j x_j \rightarrow \max$$

за умов

$$x_1 + x_2 + \dots + x_n \leq C;$$

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

Однак, портфель цінних паперів може формуватися з урахуванням різних обмежень, які пов'язані з політикою фірми. Наприклад, більшість інвестиційних фірм обмежують розміри вкладень у звичайні акції, оскільки дохід від них схильний до значних коливань. Таке обмеження можна записати у такий спосіб:

$$\sum_{j \in J_1} x_j \leq b_1,$$

де множина J_1 містить індекси різних видів звичайних акцій, а b_1 – максимально допустиме вкладення у звичайні акції.

Окрім цього, як правило, необхідно враховувати той факт, що керуючі органи багатьох інвестиційних фірм вважають за необхідне тримати певну частину капіталу у вигляді готівки або в іншій еквівалентній формі для задоволення можливих вимог вкладників (тобто у вигляді високоліквідних цінних паперів). Це обмеження записується так:

$$\sum_{j \in J_2} x_j \leq b_2,$$

де індекси з множини J_2 відповідають цінним паперам, еквівалентним готівковим коштам (наприклад, ощадним рахункам, поточним рахункам), b_2 – мінімальна величина необхідної готівки. Можна розглядати й інші обмеження такого ґатибу.

Описана модель є дуже простою задачею лінійного програмування. Головним її недоліком є неврахування ризику, пов'язаного з інвестиціями. Розрахований на основі такої моделі портфель цінних паперів може обіцяти високий середній очікуваний дохід, але при цьому ризик, пов'язаний з інвестиціями, також буде великим і справжній дохід може виявитися значно нижчим за очікуваний!

Модель II. Враховує фактор ризику, пов'язаний із кожним видом цінних паперів. Курс деяких цінних паперів, наприклад «спекулятивних акцій», має тенденцію до сильних коливань, що збільшує фактор ризику, пов'язаний з ними, але середній очікуваний дохід від них високий через їхню здатність до сильного підвищення. З іншого боку, «безпечні» інвестиції, такі, як поточні рахунки, банківські депозитні сертифікати, можуть давати менший дохід. Мірою інвестиційного ризику може бути величина відхилення доходу від середнього значення протягом останніх T років.

Позначимо через σ_{jj}^2 інвестиційний ризик для цінних паперів виду j , який розраховується як оцінка дисперсії за такою формулою:

$$\sigma_{jj}^2 = \frac{1}{T} \sum_{t=1}^T (r_j(t) - \mu_j)^2.$$

Величину σ_{ij}^2 , також можна обчислити на основі статистичних даних за минулі роки, а саме:

$$\sigma_{ij}^2 = \frac{1}{T} \sum_{t=1}^T (r_i(t) - \mu_i) * (r_j(t) - \mu_j).$$

У випадку, коли $i = j$, ця величина дорівнює дисперсії j -го виду паперів. Отже, мірою узагальненого інвестиційного ризику можна вважати таку величину:

$$V = \sum_{i=1}^n \sum_{j=1}^n \sigma_{ij}^2 x_i x_j = x^T Q x,$$

де $Q_{(n \times n)} = \{\sigma_{ij}^2\}$ – оцінка матриці коваріацій для n видів цінних паперів.

Власник цінних паперів може бути зацікавлений також у отриманні заданого середнього очікуваного доходу за умов мінімального ризику.

Відтак, відповідна оптимізаційна задача може бути сформульована таким чином:

$$Z = x^T Q x \rightarrow \min,$$

за умов

$$x_1 + x_2 + \dots + x_n \leq C;$$

$$\mu^T x \geq R;$$

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

де R – мінімальний очікуваний дохід під час вибору портфеля.

У модель можуть також бути введені додаткові, подібні до розглянутих вище обмеження, пов'язані з політикою фірми.

Зауважимо, що у цій моделі мінімізації ризику обмеження є лінійними, а цільова функція – квадратичною.

2.4. Планування роботи гідроенергетичного комплексу

Змістова постановка задачі. На рис. 1.3 зображено гідроенергетичний комплекс, що містить два водосховища і дві гідроелектростанції, кожна з яких використовує водне дзеркало. Горизонт планування роботи комплексу розбито на два періоди. Під час наповнення одного з водосховищ надлишок води видаляється через водоскид, який також захищає від повеней. Вода, що проходить через водоскид, впливає на виробництво електроенергії.

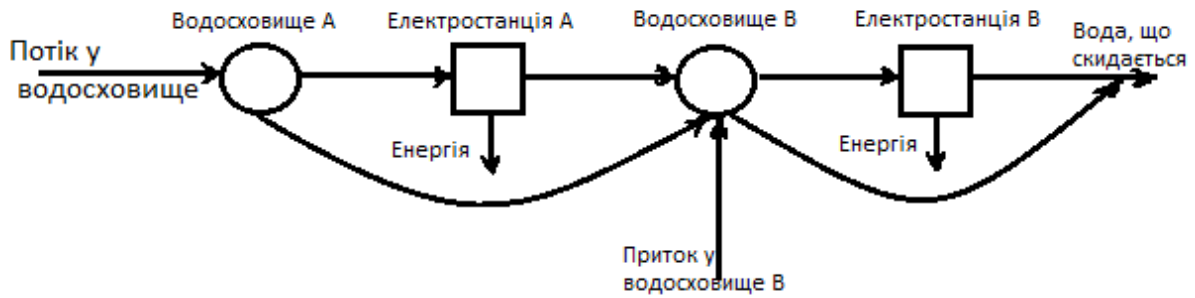


Рис.1.3. Гідроенергосистема з двох водосховищ та електростанцій

Передбачається, що один кілоакр-фут (КАФ) води на електростанції *A* перетворюється на 400 МВт·год електроенергії. Для станції *B* аналогічний показник дорівнює 200 МВт·год. Протягом одного періоду станція *A* може виробити 60 000 МВт·год електроенергії, станція *B* – 35 000 МВт·год електроенергії. Ціна 1 МВт·год електроенергії дорівнює 20 \$ за умови, що загальний обсяг її реалізації не перевищує 50 000 МВт·год. Кожен МВт·год понад 50 000 МВт·год коштує 14 \$.

Для розв'язування задачі доцільно спочатку побудувати модель функціонування системи під час першого періоду, а потім перейти до другого.

Введемо такі змінні:

- PH_1 – кількість електроенергії (у МВт·год), що продається за ціною 20 \$ за МВт·год;
- PL_1 – кількість електроенергії (у МВт·год), що продається за ціною 14 \$ за МВт·год;
- XA_1 – обсяг води (в КАФ), що спрямовується на станцію *A*;
- XB_1 – обсяг води (в КАФ), що спрямовується на станцію *B*;
- SA_1 – обсяг води (в КАФ), що скидається зі сховища *A*;
- SB_1 – обсяг води (в КАФ), що скидається зі сховища *B*;
- EA_1 – обсяг води (КАФ) в сховищі *A* в кінці періоду 1;
- EB_1 – обсяг води (КАФ) у сховищі *B* наприкінці періоду 1.

Таблиця 1.1

Вихідні дані для оптимізаційної задачі 2

	Сховище А	Сховище В
Корисний обсяг	2000	1500
Прогнозний приплив води		
період 1	200	40
період 2	130	15
Мінімально допустимий об'єм води	1200	800
Об'єм води на початок періоду 1	1900	850

Загальна кількість енергії, яка виробляється протягом періоду 1, дорівнює $400X_{A1} + 200X_{B1}$, а загальна кількість енергії для продажу складає $PH_1 + PL_1$. Отже, умова балансу виробленої електроенергії записується таким чином:

$$400X_{A1} + 200X_{B1} = PH_1 + PL_1.$$

Оскільки максимальна кількість енергії, що продається за вищою ціною, дорівнює 50 000 МВт·год, виконується нерівність $PH \leq 50\,000$. Потужність станції А, яка описується в одиницях об'єму води (КАФ), становить $60\,000/400 = 150$. Тому $X_{A1} \leq 150$. Аналогічно $X_{B1} \leq 87,5$.

Умову водного балансу для сховища А можна записати у такому вигляді:

$$\begin{array}{l} \text{Вода, що спрямовується на станцію А} \\ \quad + \\ \text{Вода, що скидається} \\ \quad + \\ \text{Кінцевий об'єм води в сховищі А} \end{array} = \begin{array}{l} \text{Початковий об'єм води в сховищі А} \\ \quad + \\ \text{Прогнозований приплив води.} \end{array}$$

Відтак,

$$X_{A1} + SA_1 + EA_1 = 1900 + 200 = 2100.$$

Позаяк ємність сховища А дорівнює 2000 КАФ, а мінімально допустимий об'єм води становить 1200 КАФ, виникають додаткові обмеження:

$$EA_1 \leq 2000, \quad EA_1 \geq 1200.$$

Аналогічно для сховища В отримаємо

$$X_{B1} + SB_1 + EB_1 = 850 + 40 + X_{A1} + SA_1,$$

$$800 \leq EA_1 \leq 1500.$$

Тепер легко побудувати модель для періоду 2, вводячи відповідні змінні: $PH_2, PL_2, X_{A2}, X_{B2}, SA_2, SB_2, EA_2, EB_2$.

Обмеження, пов'язані з виробництвом електроенергії протягом періоду 2, мають такий вигляд:

$$400X_{A2} + 200X_{B2} = PH_2 + PL_2,$$

$$PH_2 \leq 50\,000,$$

$$X_{A2} \leq 150,$$

$$X_{B2} \leq 87,5.$$

Умови водного балансу можна записати у вигляді

$$X_{A2} + SA_2 + EA_2 = EA_1 + 130,$$

$$1200 \leq EA_2 \leq 2000,$$

$$XB_2 + SB_2 + EB_2 = EB_1 + 15 + XA_2 + SA_2,$$

$$800 \leq EB_2 \leq 1500.$$

Крім наведених вище обмежень також необхідно врахувати умови невід'ємності змінних.

Критерій оптимальності – сумарний прибуток від продажу електроенергії – моделюється нелінійною цільовою функцією. Дійсно, якби ціна одного МВт·год була б сталою і не змінювалася залежно від обсягів виробленої електроенергії, то цільова функція була б лінійною вигляду px , де p – ціна на електроенергію, а x – обсяг електроенергії, що продається. У даному прикладі причина нелінійності цільової функції полягає в тому, що ціна електроенергії залежить від обсягу її виробництва. Це добре видно на графіку залежності загального доходу від обсягу проданої енергії (рис. 4). Ціна електроенергії, яка продається в рамках укладених угод, дорівнює 20 \$ за 1 МВт·год, і це вище ціни електроенергії, що реалізується не за укладеними договорами і дорівнює 14 \$ за 1 МВт·год. Функція, яка моделює цю залежність, є частково-лінійною (тобто нелінійною), оскільки вона має різний вид лінійності в областях $(0; 50\,000)$ та $(50\,000; \infty)$.

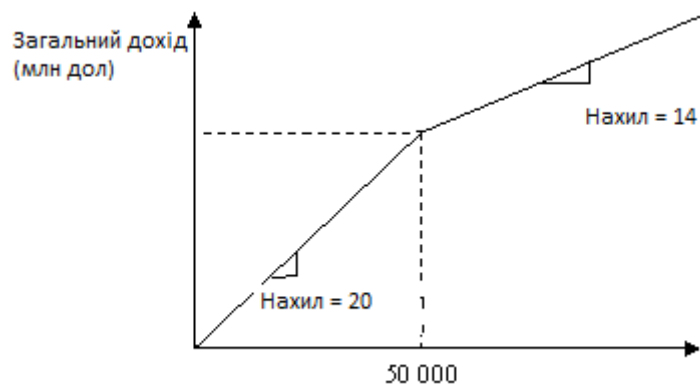


Рис. 4 Графік залежності сумарного доходу від обсягу електроенергії, що продається (МВт·год.)

Звісно, було б краще використовувати апарат лінійного програмування для формалізації цієї задачі. Цього можна досягнути штучним перетворенням, в результаті якого виходить лінійна цільова функція і лінійна модель загалом. Воно полягає в тому, що загальну кількість електроенергії, яка продається, подають у вигляді двох частин, одна з яких продається за ціною 20 \$ за 1 МВт·год, а інша – 14 \$ за 1 МВт·год, і тоді цільову функцію можна звести до лінійної.

Цей приклад показує, як у деяких випадках за допомогою нескладних перетворень можна моделювати оптимізаційні задачі, що містять нелінійні залежності методами лінійного програмування.

Максимізована функція, яка виражає загальний прибуток від продажу електроенергії, задається такою формулою:

$$Z = 20(PH_1 + PH_2) + 14(PL_1 + PL_2).$$

Вочевидь, описана модель містить 16 змінних та 20 функціональних обмежень.

Питання для обговорення

1. Скільки балансових обмежень у цій оптимізаційній моделі? Поясніть їхній сенс.

2. Як можна записати цільову функцію, якщо не використовувати подання загального обсягу електроенергії у вигляді двох змінних? Чому такий спосіб гірший у порівнянні із розглянутим вище?

3. Чи передбачено в описаній моделі умову, яка забороняє отримання рішення, що забезпечує вироблення дорогої електроенергії у незначному обсязі, меншому, ніж 50 000 МВт·год, і при цьому виробляти дешеву електроенергію?

2.5. Геометричні задачі, які зводяться до задач нелінійного програмування

Знайти конус найбільшого об'єму, який можна вписати в кулю з радіусом R . Розглянемо осьовий переріз вписаного в кулю конуса (рис. 5).

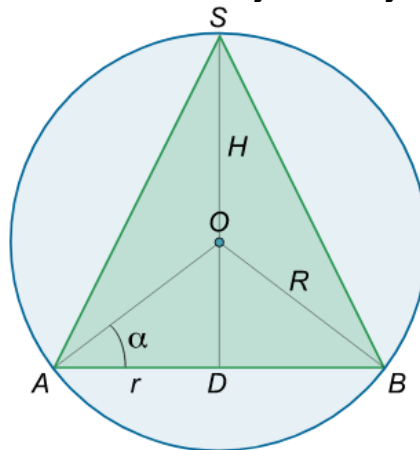


Рис. 5. Осьовий переріз вписаного в кулю конуса

Введемо такі позначення: H – висота конуса, r – радіус основи конуса, α – кут між радіусом та основою конуса. Радіус основи та висота конуса пов'язані з радіусом кулі наступними співвідношеннями:

$$r = R \cos \alpha, H = R \sin \alpha + R.$$

У такому у випадку об'єм конуса можна подати у вигляді

$$V = \frac{1}{3} \pi r^2 H = \frac{1}{3} \pi (R \cos \alpha)^2 (R \sin \alpha + R) = \frac{1}{3} \pi R^3 (\cos \alpha)^2 (\sin \alpha + 1),$$

¹ <https://calcsbox.com/usecase/49.html>

де кут α змінюється в інтервалі $0 < \alpha < \pi/2$. Диференціюємо об'єм V за змінною α :

$$\begin{aligned} V' &= \left[\frac{1}{3} \pi R^3 (\cos \alpha)^2 (\sin \alpha + 1) \right]' = \frac{1}{3} \pi R^3 [(\cos \alpha)^2 (\sin \alpha + 1)]' = \\ &= \frac{1}{3} \pi R^3 [2 \cos \alpha (-\sin \alpha) \cdot (\sin \alpha + 1) + \cos^2 \alpha \cdot \cos \alpha] = \\ &= \frac{1}{3} \pi R^3 \cos \alpha [\cos^2 \alpha - 2 \sin^2 \alpha - 2 \sin \alpha]; \\ V' = 0 &\Rightarrow \frac{1}{3} \pi R^3 \cos \alpha [\cos^2 \alpha - 2 \sin^2 \alpha - 2 \sin \alpha] = 0; \end{aligned}$$

1. $\cos \alpha = 0; \Rightarrow \alpha = \frac{\pi}{2};$
2. $\cos^2 \alpha - 2 \sin^2 \alpha - 2 \sin \alpha = 0;$
 $1 - 3 \sin^2 \alpha - 2 \sin \alpha = 0; \{ \sin \alpha = t \}$

$$3t^2 + 2t - 1 = 0;$$

$$t_{1,2} = \frac{-2 \pm \sqrt{16}}{6};$$

$$t_1 = -1, \Rightarrow \sin \alpha = -1, \Rightarrow \alpha = \frac{3\pi}{2};$$

$$t_2 = \frac{1}{3}, \Rightarrow \sin \alpha = \frac{1}{3}.$$

Зрозуміло, що розв'язком задачі є $\sin \alpha = \frac{1}{3}$.

Можна впевнитися, що при зростанні кута α і переході через цю точку похідна змінює свій знак із плюса на мінус, тобто таке значення синусу кута забезпечує максимальне значення об'єму конуса.

Обчислимо косинус кута α :

$$\cos \alpha = \sqrt{1 - \left(\frac{1}{3}\right)^2} = \frac{2\sqrt{2}}{3}.$$

Тоді радіус основи та висота конуса найбільшого об'єму мають такі значення:

$$r = \frac{2\sqrt{2}}{3} R, H = R \cdot \frac{1}{3} + R = \frac{4}{3} R,$$

а його об'єм

$$V = \frac{1}{3} \pi r^2 H = \frac{\pi}{3} \cdot \left(\frac{2\sqrt{2}}{3} R\right)^2 \cdot \frac{4}{3} R = \frac{32\pi}{81} \cdot R^3,$$

що становить $8/27$ об'єму кулі.

Рекомендована література та інформаційні ресурси

1. Математичні методи дослідження операцій: підручник / Є.А. Лавров, Л.П. Пертун, В.В. Шендрік та ін. – Суми : Сумський державний університет, 2017. – 212 с.
2. Нефьодов Ю.М. Методи оптимізації в прикладах і задачах : навчальний посібник / Ю.М. Нефьодов, Т.Ю. Балицька. – Київ : Кондор, 2011. – 324 с.
3. Бех О.В. Математичне програмування : навч. посібн. / О.В. Бех, Т.А. Городня, А.Ф. Щербак. – Львів : Магнолія-2006, 2014. – 200 с.
4. Штовба С.Д. Методи оптимізації в середовищі MatLab. Лабораторний практикум. Навчальний посібник. – Вінниця: ВДТУ, 2001. – 56с.
5. <http://www.mathnet.ru/links/d15bf9d68ce568124f309ee5b2c4e2d4/mo90.pdf>]

Завдання для розв'язування у класі

1. У множині всіх трикутників з заданим периметром $2p$ знайти трикутник з найбільшою площею.
2. На множині усіх чотирикутників, вписаних у коло радіусом R , знайти чотирикутник, який має найбільшу площу.
3. Нехай r і R – радіуси вписаного та описаного кіл трикутника відповідно. Визначити найбільше значення відношення r/R .
4. Серед усіх прямокутних трикутників із заданою гіпотенузою знайти той, що має найбільший радіус вписаного кола r .
5. Знайти всередині трикутника точку, сума квадратів відстаней від якої до вершин трикутника була б найменшою.
6. З усіх прямокутних паралелепіпедів з даною діагоналлю d знайти той, який має найбільшу площу повної поверхні S .
7. З кола радіусом R вирізано сектор з кутом a при вершині, і частину, яка залишилася, згорнуто у конічну воронку. При якому куті a отримуємо воронку максимальної місткості?
8. У кулю радіусом R вписати конус, що має найбільшу площу повної поверхні S .
9. Серед усіх перерізів одиничного куба, які проходять через його діагональ, знайти переріз найменшої площі й саму цю площу.

Контрольні запитання

1. Сформулюйте математичну постановку задачі нелінійного програмування?
2. Назвіть основні етапи побудови математичних моделей оптимізаційних задач?
3. Які задачі оптимізації складають клас сепарабельного програмування?
4. Які задачі складають клас динамічного програмування?
5. Наведіть приклад задачі сепарабельного програмування.
6. Чим відрізняються задачі умовної і безумовної оптимізації?
7. Як формулюється задача опуклого програмування?

Завдання для самостійної та індивідуальної роботи

Побудувати математичні моделі задач відповідно до варіанта (див. таблицю нижче), реалізувати їх за допомогою надбудови «Розв'язувач» Microsoft Excel. Проаналізувати отримані розв'язки і уточнити моделі, якщо це необхідно.

Таблиця 1.2

Варіанти завдань

Номер варіанта	Номери задач за розділами					
	Планіметрія			Стереометрія		Виробничі завдання
1	19	21	6	3	4	3
2	14	20	14	3	3	1
3	25	5	3	5	2	5
4	17	21	11	8	10	8
5	29	2	5	2	8	2
6	18	16	10	7	9	7
7	9	25	13	4	2	4
8	27	18	6	1	9	1
9	1	5	9	10	1	9
10	21	13	19	4	7	4
11	3	25	13	10	8	2
12	24	5	17	8	6	8
13	8	20	21	5	7	5
14	10	6	11	7	4	7
15	2	1	24	3	5	3
16	13	5	1	1	1	1
17	30	12	21	9	6	9
18	15	4	7	7	8	7
19	12	19	21	10	3	7
20	26	17	11	6	7	6
21	5	5	14	4	4	4
22	22	19	10	9	7	9
23	7	23	7	8	10	8
24	28	17	10	4	3	4
25	16	10	9	6	9	6
26	11	24	13	10	1	3
27	23	20	22	1	6	1
28	20	25	15	2	10	2
29	4	18	6	5	5	5
30	6	2	7	6	2	6

Завдання із планіметрії

1. У задане коло вписати прямокутник найбільшої площі.
2. У задане півколо вписати прямокутник найбільшої площі.
3. Зі всіх прямокутників заданого периметра знайти той, що має найбільшу площу.
4. Серед усіх трикутників з певною основою та заданим кутом при вершині знайти той, периметр якого найбільший.
5. З усіх кругових секторів заданого периметра p знайти той, що має найбільшу площу S .
6. Площа трапеції дорівнює 1. Яку найменшу величину може мати більша діагональ цієї трапеції?
7. Дано два кола, які перетинаються, і мають радіуси r і R . Довжина їхньої загальної хорди дорівнює $2a$. Знайти довжину найбільшої січної, яка проходить через точку перетину кіл.
8. На стороні трикутника знайти точку, добуток відстаней від якої до двох інших сторін трикутника був би найбільшим.
9. Яким має бути кут при основі рівнобедреного трикутника заданої площі, щоб радіус вписаного в цей трикутник кола був найбільшим?
10. Помістити всередині квадрата зі стороною a рівносторонній трикутник найбільшого розміру. Знайти довжину сторони цього трикутника.
11. На колі радіусом R дано дві точки A та B , відстань між якими дорівнює одиниці. Яке найбільше значення може набувати сума $AC^2 + BC^2$, якщо точка C також лежить на цьому колі?
12. У коло радіуса R вписано кут a . Якими мають бути довжини хорд, які утворюють цей кут, щоб їхня сума була найбільшою?
13. З усіх чотирикутників з одними й тими самими сторонами знайти чотирикутник найбільшої площі.
14. Дано площу S і кут a трикутника. Знайти мінімум: а) суми двох сторін, яку утворюють цей кут; б) сторони, що протилежна даному куту; в) периметра трикутника.
15. З усіх трикутників, вписаних у коло, і які мають спільну основу, знайти трикутник із найменшою площею.
16. З усіх трикутників, вписаних у коло, знайти трикутник із найбільшою площею.
17. Бічні сторони та одна з основ трапеції дорівнюють a . Якою має бути довжина іншої основи, аби трапеція мала найбільшу площу?
18. Знайти основу рівнобедреного трикутника з стороною b , в який вписано коло найбільшого радіусу.
19. У коло радіусом R вписаний рівнобедрений трикутник. Яке найбільше значення може набувати його висота, проведена до бічної сторони, і яким буде значення кута при вершині?
20. Бісектриса гострого кута прямокутного трикутника дорівнює одиниці. Яке найбільше значення може набувати висота трикутника, проведена до гіпотенузи?

21. Вирізати із квадрата шестикутник найбільшої площі.
22. Знайти довжину найменшого відрізка, котрий розділяє рівносторонній трикутник зі стороною a на дві рівновеликі частини.
23. Яка найбільша площа прямокутника, вписаного в ромб із діагоналями $2a$ та $2b$?
24. Довести, що з усіх трикутників з даною стороною $BC = a$ та даним протилежним кутом A рівнобедрений трикутник має найменшу медіану, якщо кут A при вершині тупий, і найбільшу, якщо кут A гострий.
25. У трикутнику задано сторона a та периметр $2p$. Які довжини повинні мати дві інші сторони цього трикутника, аби його площа була максимальною?

Завдання із стереометрії

1. Біля кулі радіусом R описати конус найменшого об'єму V .
2. Основою прямої призми є прямокутний трикутник з площа якого дорівнює 2. Висота призми дорівнює гіпотенузі основи. Якими мають бути сторони основи, аби бічна поверхня призми була найменшою?
3. У кулю вписано конус найбільшого об'єму, а в нього – циліндр найбільшого об'єму. Знайдіть відношення висоти циліндра до радіуса кулі.
4. Вписати в задану кулю радіуса r циліндр з найбільшою бічною поверхнею.
5. Колода довжиною 20 дм має форму усіченого конуса з діаметрами основ 2 дм та 1 дм. Потрібно вирубати з колоди брус із квадратним поперечним перерізом, вісь якого збігалася б з віссю колоди та об'єм якого був би найбільшим. Як це зробити?
6. З усіх паралелепіпедів з певною сумою S трьох взаємно перпендикулярних ребер знайти той, об'єм якого найбільший.
7. У правильній трикутній піраміді сума квадратів довжин всіх ребер дорівнює Q . Яке найбільше значення може мати її бічна поверхня S ?
8. Радіус основи прямого кругового конуса дорівнює R , висота становить H . У конус вписано циліндр, при цьому одна його основа лежить на основі конуса, а інша – на бічній поверхні. Яке найбільше значення може мати бічна поверхня циліндра?
9. Основою прямої призми є прямокутний трикутник. Сума довжин всіх її ребер дорівнює 1. Знайти найбільше значення бічної поверхні S .
10. Потрібно вирубати з граніту постамент у формі прямокутного паралелепіпеда, висота якого дорівнює діагоналі основи, а площа основи складає: а) 4 м; б) 2 м. Якими мають бути вимірювання паралелепіпеда, аби об'єм постаменту виявився найменшим?

Виробничі завдання

1. Чотири підприємства можуть виробляти продукцію в основний час у кількості 160, 250, 390, 200 одиниць та додатково, використовуючи понаднормовий час, у кількостях 40, 50, 110, 50 од. При цьому витрати на

виробництво одиниці продукції на підприємствах становлять 30, 50, 40, 35 – у основний час і на 50 % вище при роботі у понаднормовий. Вироблена продукція має бути доставлена чотирьом споживачам у кількостях 380, 250, 400 та 200 од. відповідно.

Транспортні витрати c_{ik} на перевезення одиниці виробленої продукції від i -го підприємства ($i = 1,2,3,4$) k -му споживачеві ($k = 1,2,3,4$) задано матрицею

$$C = \begin{pmatrix} 25 & 30 & 40 & 35 \\ 30 & 40 & 30 & 20 \\ 25 & 45 & 30 & 40 \\ 42 & 36 & 28 & 34 \end{pmatrix}.$$

Визначити оптимальну виробничу програму для кожного підприємства та оптимальний план перевезень продукції до споживачів.

Вказівка. Крім звичайних змінних x_{ik} для позначення перевезень від i -го підприємства до k -го споживача, слід ввести додаткові змінні x_i , які описують сумарний випуск продукції i -м підприємством.

2. Визначити оптимальне розподілення загальної суми капіталовкладень, яка становить 120 млн грн, для будівництва трьох підприємств. При цьому величина прибутку (у млн грн) на кінець року на кожен млн грн капіталовкладень залежить не тільки від підприємства, а також від величини виділених йому коштів, спадаючи зі зростанням останніх (у зв'язку з труднощами їх освоєння). Ці залежності подано у табл. 1.3

Таблиця 1.3

Вихідні дані задачі 2

Підприємство	Обсяг капітальних вкладень	Прибуток	Обсяг капітальних вкладень	Прибуток	Обсяг капітальних вкладень	Прибуток
I	до 15	1,5	15 – 30	0,5	більше 30	0
II	до 20	1,0	20 – 40	0,4	40 – 50	0,2
III	до 10	1,2	10 – 15	1,0	15 – 30	0,5

3. Визначити місце будівництва заводу між двома пунктами збуту, відстань між якими становить 100 км, та розмір поставок у кожен пункт, якщо валовий випуск продукції заводу складає 150 одиниць, а залежність продажної ціни одиниці продукції від кількості продукції x_i ($i = 1,2$), що поставляється у кожен з пунктів збуту та витрат на перевезення одиниці продукції від відстані u_i між заводом та пунктом збуту, показано у табл. 1.4.

Таблиця 1.4

Вихідні дані задачі 3

Пункти збуту	Продажна ціна, грн./од.	Витрати, грн/од.
1	$15 - 0,1x_1$	$1,5 + 0,1y_1$
2	$12 - 0,08x_2$	$1,5 + 0,05y_2$

4. Підприємство може випускати три види продукції A , B і B , маючи для цього ресурси сировини обсягом 1000 одиниць, яка витрачається в кількості 5,

2,5 і 2 од. відповідно на кожну одиницю продукції A , B і B . Прибуток, одержуваний підприємством на кожну одиницю продукції, залежить від її виду та від загальної величини ресурсів, які витрачено на виробництво, згідно з даними табл. 1.5. Необхідно визначити оптимальний план випуску продукції.

Таблиця 1.5

Вихідні дані задачі 4

Витрати ресурсів	А		Б		В	
	До 100	100-200	До 200	200-400	До 300	300-600
Прибуток	5	3	10	6	4	2

5. Випуск продукції може здійснюватися за двома технологічними режимами. Витрати на виробництво для кожного з них дорівнюють $21x_1^{4/3}$ та $4x_2^{3/2}$ відповідно, де x_1 та x_2 – обсяг продукції, що виробляється відповідно до 1-го та 2-го технологічного режиму. При цьому другий технологічний режим передбачає використання привезеної сировини, і тому пов'язаний з додатковими витратами на транспортування, які складають $400x_2^{1/2}$. Визначити оптимальний план використання технологічних режимів, який забезпечить загальний випуск продукції в кількості 100 од.

6. Визначити оптимальний розмір партії сировини, яка закуповується через різні проміжки часу, якщо річна потреба в ньому становить Q од., витрати сировини рівномірні, річні витрати на зберігання одиниці сировини дорівнюють α , а витрати на закупівлю нової партії – β .

Вказівка. Якщо партія розміром x витрачається рівномірно протягом року, то середньорічний запас сировини дорівнює $\frac{x}{2}$.

7. Розв'язати попередню задачу за умови, що місткість складу для зберігання запасів не перевищує D .

8. Визначити оптимальну програму випуску двох видів продукції з урахуванням обмежених ресурсів сировини (120 кг), обладнання (300 верстатогод) та електроенергії (280 кВт*год), враховуючи, що встановлено такі норми витрат на одиницю продукції: сировини – 3 та 2 кг/од., електроенергії – 4 і 7 кВт*год та обладнання – $(50 - 5x_1)$ і $(20 - 4x_2)$, де x_1 та x_2 – кількість вироблених одиниць продукції 1-го і 2-го виду відповідно.

9. Розглянемо тригалузеву модель економіки з технологічною матрицею $\|a_{ik}\|$ ($i = 1,2,3$, й $k = 1,2,3$), де a_{ik} – коефіцієнт прямих витрат. Задано коефіцієнти прямих витрат праці (t_k) та капіталовкладень (c_k), загальний обсяг трудових ресурсів T , потреби в кінцевому продукті (b_i), ціни на продукти, які імпортуються (a_i) та на експортовані продукти $\beta_i = \gamma_i - \delta_i u_i$, де u_i – розмір експорту i -го продукту та максимально допустимий дефіцит зовнішньої торгівлі D (тобто різниця між сумарною вартістю імпорту та експорту).

Скласти математичну модель для визначення валового випуску продукції, обсягу експорту та імпорту, які потребують мінімум капіталовкладень.

Практичне заняття № 2

Графічний метод розв'язання нелінійного програмування

Мета: засвоєння теоретичного матеріалу стосовно моделей та методів нелінійного програмування, здобуття практичних навичок розв'язання задач НЛП за допомогою графічного способу. Вивчення властивостей задач НЛП.

План заняття

1. Постановка задачі НЛП з двома змінними;
2. Графічний метод розв'язання задачі НЛП;
3. Приклади розв'язування задач.

Опорні поняття

1. Постановка задачі НЛП з двома змінними

Загальну постановку задачі нелінійного програмування із двома змінними можна записати у такий спосіб:

$$f(x_1, x_2) \rightarrow \min(\max), (x_1, x_2) \in \Omega,$$

$$\Omega = \{(x_1, x_2) \in D: g_i(x_1, x_2) \leq 0, \quad 1 \leq i \leq m; \quad g_i(x_1, x_2) = 0, \quad m + 1 \leq i \leq s\},$$

$$D = \{(x_1, x_2) \in \mathbb{R}^2: a_j \leq x_j \leq b_j, \quad 1 \leq j \leq 2\}.$$

2. Графічний метод розв'язання задачі НЛП

Коли задача нелінійного програмування має тільки дві змінні, її можна розв'язати графічно. Опишемо схему пошуку розв'язку.

1. Знаходимо область допустимих розв'язків задачі Ω (якщо вона порожня, то задача не має розв'язку);

2. Будуємо гіперповерхню $f(x_1, x_2) = C$;

3. Визначаємо гіперповерхню найнижчого (найвищого) рівня або встановлюємо неможливість розв'язання задачі через необмеженість функції знизу (зверху) на множині допустимих розв'язків;

4. Знаходимо точку області допустимих розв'язків, через яку проходить гіперповерхня найнижчого (найвищого) рівня, та обчислюємо в ній значення функції f .

Точки, які задовольняють систему обмежень, називають допустимими розв'язками задачі нелінійного програмування. Вони утворюють *множину допустимих розв'язків* (МДР). Допустимий розв'язок, якому відповідає екстремальне значення цільової функції, називають *оптимальним* для задачі нелінійного програмування і позначають x_{opt} .

3. Ілюстративні приклади

Приклад 2.1. Знайти значення змінних x_1, x_2 , за яких функція

$$z(x) = (x_1 - 2)^2 + (x_2 - 7)^2$$

набуває екстремального значення, враховуючи такі умови:

$$\begin{cases} -x_1 + x_2 \leq 3, \\ 5x_1 + 3x_2 \leq 97, \\ x_1 + 7x_2 \geq 74, \end{cases}$$

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

Розв'язання. Введемо на площині прямокутну систему координат $x_1 O x_2$.

1. Побудуємо множину допустимих розв'язків, яка визначається умовами задачі. Для цього потрібно побудувати граничні прямі, а саме:

$$-x_1 + x_2 = 3, \quad (2.1)$$

$$5x_1 + 3x_2 = 97, \quad (2.2)$$

$$x_1 + 7x_2 = 74, \quad (2.3)$$

(наприклад, за точками їх перетину з осями координат), а далі визначити відповідні півплощини.

Взагалі, аби визначити, яку півплощину описує нерівність $a_1 x_1 + a_2 x_2 \leq b$, (знак нерівності вибраний довільно), потрібно:

1) побудувати пряму $a_1 x_1 + a_2 x_2 = b$;

2) визначити, з якої сторони від неї розміщена точка $O(0; 0)$ (або будь-яка інша, що не лежить на даній прямій):

- якщо координати обраної точки задовольняють дану нерівність, то півплощина, в якій знаходиться обрана точка, є шуканою,
- якщо координати обраної точки не задовольняють дану нерівність, то шуканою півплощиною буде та, яка цю точку не містить.

Наприклад, опишемо побудову півплощини для першого обмеження задачі більш детально. Цьому обмеженню відповідає пряма (1). Знаходимо точки її перетину з осями Ox_1 і Ox_2 – відповідно $(-3; 0)$ та $(0; 3)$. Через них проводимо пряму (1). Далі підставляємо координати будь-якої точки площини в нерівність $-x_1 + x_2 \leq 3$, приміром, за таку точку можна взяти початок координат $(0; 0)$. Тоді $-0 + 0 \leq 3$. Оскільки отримали правильну нерівність, геометричним місцем точок, координати яких задовольняють нерівність $-x_1 + x_2 \leq 3$, є півплощина, що містить точку $O(0; 0)$ (рис. 2.1).

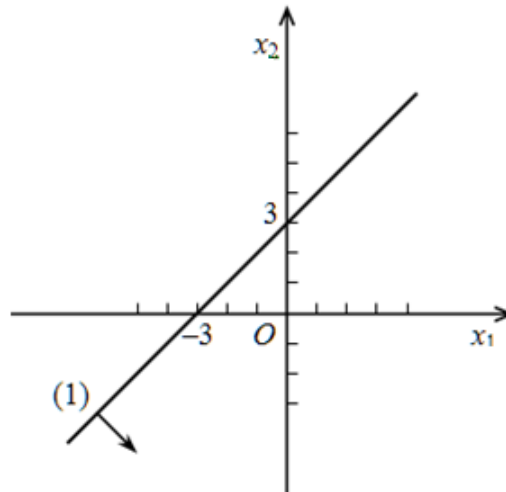


Рис. 2.1. Побудова півплощини, що задається лінійною нерівністю

Множина точок, координати яких задовольняють решті нерівностей, зображена на рис. 2.2.

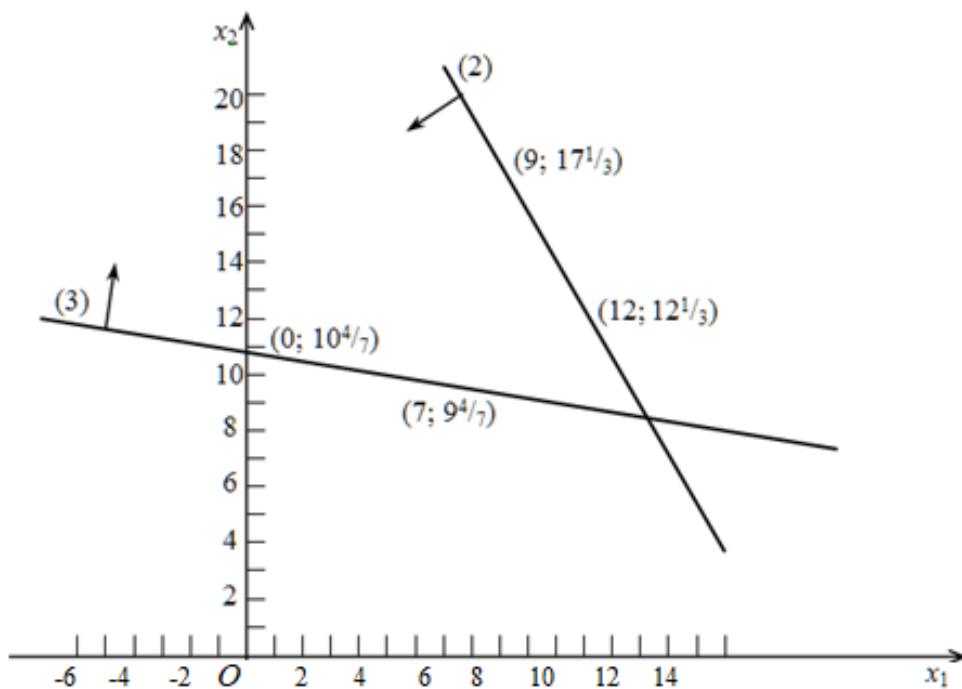


Рис. 2.2. Множина розв'язків системи двох лінійних нерівностей

З умов $x_1 \geq 0$, $x_2 \geq 0$ випливає, що геометричний образ МДР має розташовуватися в першій чверті площини $x_1 O x_2$. Перетином усіх отриманих півплощин є зображений на рис. 2.3. трикутник ABD , який і є множиною допустимих розв'язків.

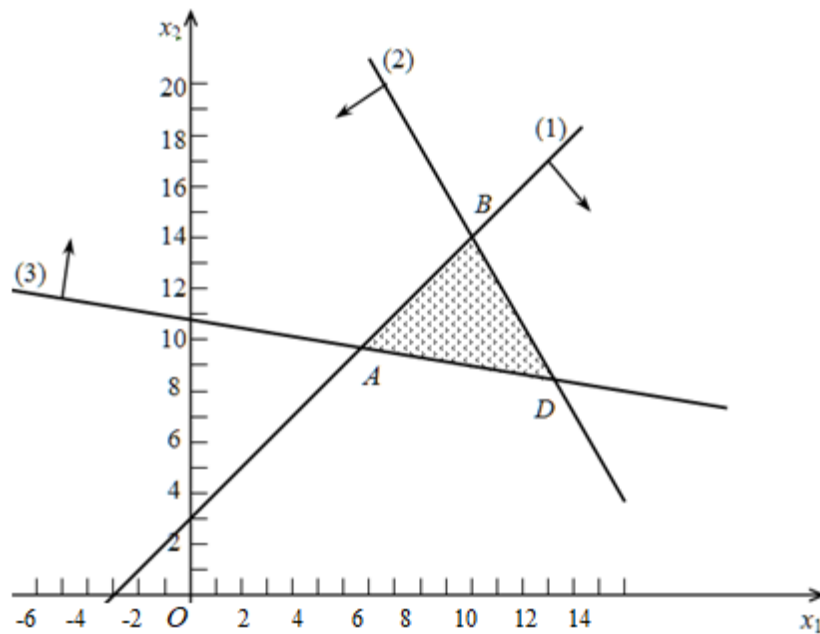


Рис. 2.3. Множина допустимих розв'язків

Тепер будемо лінії рівня цільової функції відповідно до рівняння $(x_1 - 2)^2 + (x_2 - 7)^2 = Const$ (рис. 9) і різних значень константи. Вочевидь, вони являють собою кола з центром в точці $(2, 7)$, яка є точкою безумовного мінімуму функції. Зі зростанням радіусу кола збільшується і значення цільової функції на ньому. На рис. 2.4 зображено лінії рівня функції. Ті, на яких вона набуває найменшого і найбільшого значення на допустимій множині розв'язків $z = z_{min}$ та $z = z_{max}$, виділено товстою лінією.

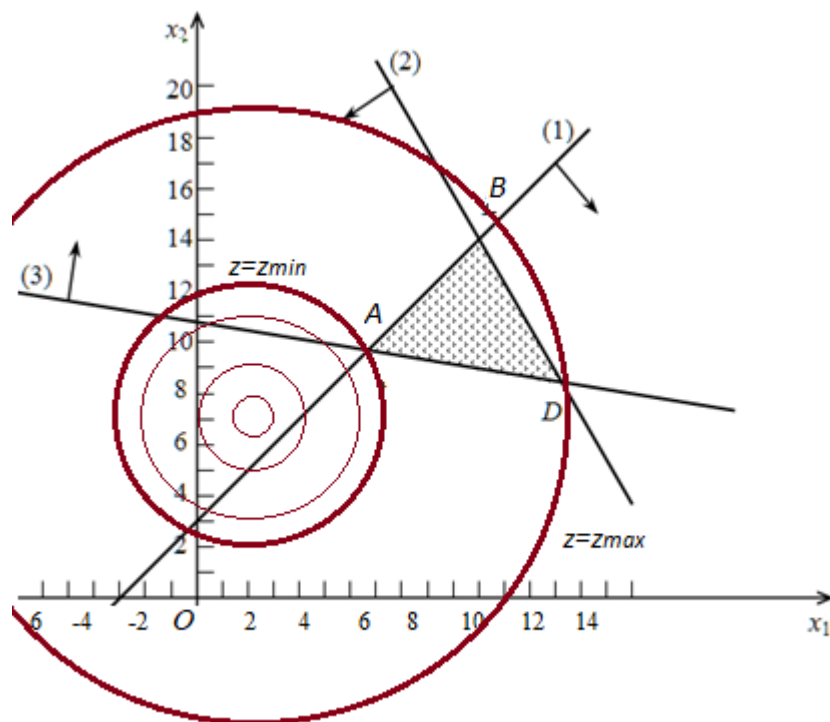


Рис. 2.4. Графічне розв'язання прикладу 1

Знайдемо значення змінних x_1 , x_2 , за яких цільова функція приймає максимальне і мінімальне значення.

Максимального значення функція набуває в точці D (точці виходу з МДР), координати якої визначаються як перетин двох прямих (3) і (2). Її координати визначаємо, розв'язуючи таку систему:

$$\begin{cases} x_1 + 7x_2 = 74, \\ 5x_1 + 3x_2 = 97. \end{cases}$$

Отже, $x_1 = 14,28125$; $x_2 = 8,53125$. Обчислимо значення максимальне цільової функції: $z_{\max} = (14,28125 - 2)^2 + (8,53125 - 7)^2 = 153,1738$.

Тепер знайдемо значення змінних x_1 , x_2 , за яких цільова функція мінімізується – це координати точки A – точки перетину прямих (1) та (3). Розв'язуючи систему

$$\begin{cases} x_1 + x_2 = 3, \\ x_1 + 7x_2 = 74, \end{cases}$$

знаходимо координати точки $A(6,625; 9,625)$. Мінімальне значення цільової функції $z_{\min} = z(A) = 28,28125$.

Приклад 2.2. Розв'язати подану нижче екстремальну задачу.

$$f = -(x_1 - 4)^2 - 4(x_2 - 2)^2 \rightarrow \text{extr}$$

$$\begin{cases} 2x_1 + 5x_2 \geq 10, \\ x_1 - 3x_2 \leq 3,5, \\ -x_1 + x_2 \leq 1, \\ x_1, x_2 \geq 0, \text{ цілі.} \end{cases}$$

Розв'язання. Введемо на площині прямокутну систему координат $x_1 O x_2$.

Множина допустимих значень для цієї задачі – необмежена багатокутна. Її зображено на рис. 10.

Лініями рівня цільової функції $(x_1 - 4)^2 + 4(x_2 - 2)^2 = C$ є еліпси з центром в точці $P(4, 2)$, у якого більша піввісь вдвічі перевищує меншу. Цільова функція спадає по мірі збільшення обхвату відповідного еліпса. Відтак, максимальне значення вона набуває в точці P , а мінімального значення у функції на заданій множині немає. На рис. 10 видно, що $f_{\min} = -\infty$, тобто задача на мінімум є нерозв'язною, а $f_{\max} = 0$.

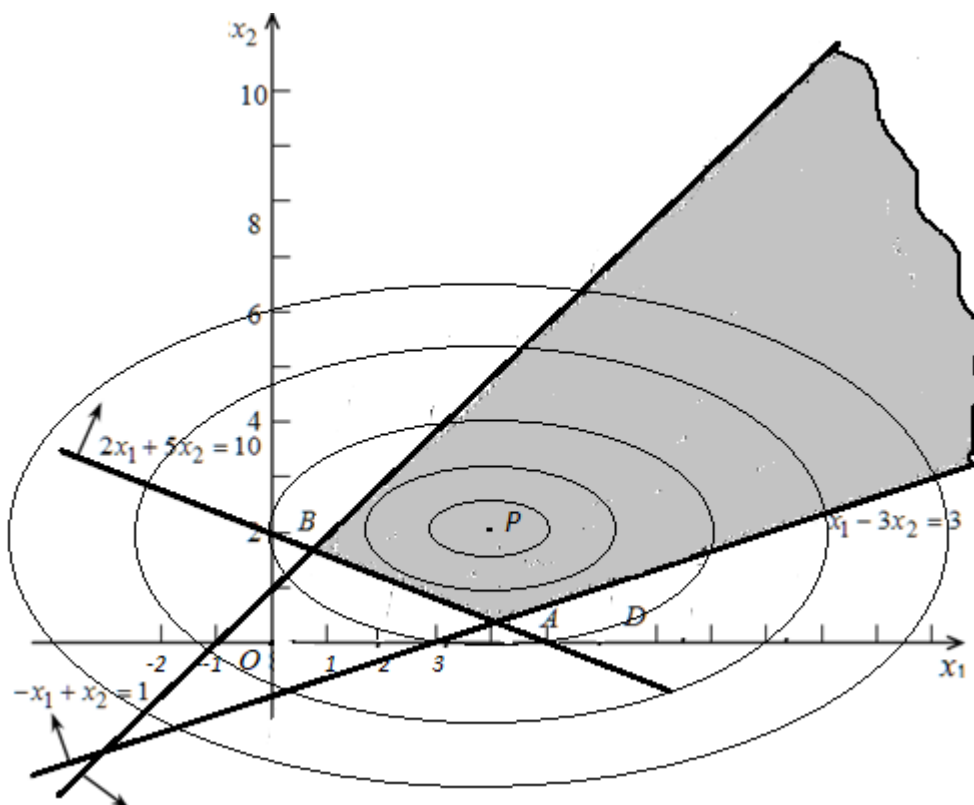


Рис. 10. Необмеженість МДР і цільової функції

Рекомендована література та інформаційні ресурси

1. Ржевський С.В. Дослідження операцій / С.В. Ржевський, В.М.Александрова. – К.: Академвидав, 2006. – 560 с.
2. Математичні методи дослідження операцій: підручник / Є.А. Лавров, Л.П. Пертун, В.В. Шендрик та ін. – Суми : Сумський державний університет, 2017. – 212 с.
3. Нефьодов Ю.М. Методи оптимізації в прикладах і задачах : навчальний посібник / Ю.М. Нефьодов, Т.Ю. Балицька. – Київ : Кондор, 2011. – 324 с.
4. Бех О.В. Математичне програмування : навч. посібн. / О.В. Бех, Т.А. Городня, А.Ф. Щербак. – Львів : Магнолія–2006, 2014. – 200 с.

Завдання для розв’язування у класі

Розв’язати подані нижче задачі НЛП графічним методом.

- | | | |
|--|--|--|
| <p>1. $L = 1/x_1 \rightarrow \text{extr}$
 $x_1 + x_2 \leq 1,$
 $-x_1 + x_2 \geq -1,$
 $x_1 \geq 0, x_2 \geq 0.$</p> | <p>2. $(x_1 + 1)^2 + 4(x_2)^2 \rightarrow \text{extr}$
 $-4x_1 + 5x_2 \leq 20,$
 $2x_1 + x_2 \geq 6,$
 $5x_1 - x_2 \leq 45,$
 $x_1 - x_2 \leq 6, x_1 \geq 0, x_2 \geq 0.$</p> | <p>3. $(x_1 - 5)^2 - 2x_2 \rightarrow \text{extr}$
 $x_1 - x_2 \leq 1,$
 $x_1 + x_2 \geq 2,$
 $x_1 - 2x_2 \leq 0,$
 $x_1 \geq 0, x_2 \geq 0.$</p> |
|--|--|--|

4. $L = 2x_1 + 3x_2 \rightarrow \text{extr}$	5. $L = 2x_1 + 3x_2 \rightarrow \text{extr}$	6. $x_1 + 3/(x_2-1) \rightarrow \text{extr}$
$(x_1-2)^2 - 2x_2 \leq -6,$	$(x_1-2)^2 + (x_2-2)^2 \leq 4,$	$x_1 - x_2 \leq 1,$
$x_1 + 2x_2 \geq 8,$	$(x_1-2)^2 + (x_2-2)^2 \geq 1,$	$x_1 - x_2 \geq 0,$
$x_1 \geq 0.$		$2x_1 + x_2 \leq 2.$

Контрольні запитання

- Сформулюйте задачу нелінійного програмування. Наведіть приклади.
- У яких випадках задача НЛП не має розв'язків?
- Дайте визначення лінії рівня функції? Який вигляд вона має для лінійної функції двох змінних? Якого вигляду вона набуває для нелінійної функції двох змінних? Наведіть приклади.
- Чи може задача нелінійного програмування мати безліч розв'язків? Якщо так, то наведіть приклад такої задачі.
- Наведіть приклад нерозв'язної задачі математичного програмування.
- Яка множина на площині відповідає обмеженню лінійної нерівності?

Завдання для самостійної та індивідуальної роботи

Розв'язати задачі нелінійного програмування графічним методом, використовуючи набуті знання з математичного аналізу, алгебри та аналітичної геометрії. Кожен студент розв'язує дві задачі. Остання цифра номера студента в журналі академічної групи має збігатися з останньою цифрою номера задачі.

1	$f = (x_1 - 2)^2 + (x_2 - 2)^2 \rightarrow \text{extr}$	2	$z = 2x^2 + 2y^2 - x + 3y + 6 \rightarrow \text{extr}$
	$\begin{cases} 3x_1 + 4x_2 \leq 30 \\ x_1 x_2 \geq 8 \end{cases}$		$\begin{cases} -4x + 8y \leq 32; \\ 6x + 2y = 12. \end{cases}$
3	$f = (x_1 - 5)^2 + (x_2 - 2)^2 \rightarrow \text{extr}$	4	$f = (x_1 - 1)^2 + (x_2 - 1)^2 \rightarrow \text{extr}$
	$\begin{cases} 3x_1 + 2x_2 \geq 7 \\ 8x_1 - x_2 \leq 8 \\ -9x_1 + 2x_2 \leq 6 \\ x_1 \geq 0, x_2 \geq 0 \end{cases}$		$\begin{cases} 2x_1 + x_2 \leq 10 \\ -2x_1 + 3x_2 \leq 6 \\ x_1 + 2x_2 \geq 4 \\ x_1 \geq 0, x_2 \geq 0 \end{cases}$
5	$x_1^2 - 4x_2^2 \rightarrow \text{extr}$	6	$-4x + y \rightarrow \text{extr}$
	$\begin{cases} x_1 - x_2 = 4, \\ x_1^2 + x_2^2 \leq 16 \end{cases}$		$\begin{cases} (x_1 - 2)^2 + (x_2 + 3)^2 = 9 \\ 0 \leq x_1 \leq 6, -4 \leq x_2 \leq 2, \end{cases}$
7	$z = 9x_1^2 + x_2^2 - 27x_1 + 2x_2 \rightarrow \text{extr},$	8	$z = 9x_1^2 + x_2^2 + 4x_1 + 36x_2 \rightarrow \text{extr},$
	$\begin{cases} 4x_1 + 5x_2 \leq 32; \\ x_1 + 4x_2 \leq 16; \\ x_j \geq 0, j = \overline{1,2}. \end{cases}$		$\begin{cases} 4x_1 + 3x_2 \leq 24; \\ 3x_1 + 2x_2 \leq 35; \\ x_j \geq 0, j = \overline{1,2}. \end{cases}$

- 9 $z = -x_1^2 - 4x_2^2 + 12x_1 + 32x_2 \rightarrow extr$ 10 $f = -(x_1 - 4)^2 - (x_2 - 4)^2 \rightarrow extr$

$$\begin{cases} 2x_1 + 3x_2 \leq 30; \\ 3x_1 + x_2 \leq 45; \\ x_j \geq 0, j = \overline{1,2}. \end{cases}$$

$$\begin{cases} (x_1 - 5)^2 + (x_2 - 5)^2 \leq 8 \\ x_j \geq 0, j = \overline{1,2} \end{cases}$$
- 11 $f = (x_1 - 6)^2 + (x_2 - 4)^2 \rightarrow extr$ 12 $f = (x_1 + 2)^2 + (x_2 - 6)^2 \rightarrow extr$

$$\begin{aligned} x_1 - x_2 &\leq 4 \\ 2x_1 - x_2 &\leq 12 \\ x_1 &\leq 4; x_1 \geq 0, x_2 \geq 0 \end{aligned}$$

$$\begin{aligned} (x_1 - 8)^2 + (x_2 - 2)^2 &\leq 9 \end{aligned}$$
- 13 $f = -(x_1 + 1)^2 - (x_2 - 5)^2 \rightarrow extr$ 14 $z = -x_1^2 - 2x_2^2 + 2x_1 + 12x_2 \rightarrow extr$

$$\begin{aligned} (x_1 + 3)^2 + (x_2 - 2)^2 &\leq 16 \\ x_j &\geq 0, j = \overline{1,2} \end{aligned}$$

$$\begin{cases} x_1 + 2x_2 \leq 8; \\ 2x_1 - x_2 \leq 12; \\ x_j \geq 0, j = \overline{1,2}. \end{cases}$$
- 15 $z = x_1^2 - x_2^2 + 12x_1 + 32x_2 \rightarrow extr$ 16 $z = -x_1^2 - 4x_2^2 + 12x_1 + 32x_2 \rightarrow extr$

$$\begin{cases} -4x_1 + 9x_2 \leq 36; \\ 8x_1 + x_2 \leq 48; \\ x_j \geq 0, j = \overline{1,2}. \end{cases}$$

$$\begin{cases} 2x_1 + 3x_2 \leq 30; \\ 3x_1 + x_2 \leq 45; \\ x_j \geq 0, j = \overline{1,2}. \end{cases}$$
- 17 $z = 4x_1^2 - x_2^2 + 16x_1 + 6x_2 \rightarrow extr$ 18 $f = (x_1 + 2)^2 + (x_2 - 6)^2 \rightarrow extr$

$$\begin{cases} 3x_1 + 2x_2 \leq 24; \\ x_1 + 3x_2 \leq 18; \\ x_j \geq 0, j = \overline{1,2}. \end{cases}$$

$$\begin{aligned} (x_1 - 8)^2 + (x_2 - 2)^2 &\leq 9 \\ x_j &\geq 0, j = \overline{1,2} \end{aligned}$$
- 19 $f = (x_1 - 5)^2 + (x_2 + 4)^2 \rightarrow extr$ 20 $z = -4x_1^2 + x_2^2 + 12x_1 - 32x_2 \rightarrow extr$

$$\begin{aligned} (x_1 - 1)^2 + (x_2 + 2)^2 &\leq 4 \\ x_j &\geq 0, j = \overline{1,2} \end{aligned}$$

$$\begin{cases} 2x_1 + 3x_2 \leq 24; \\ 3x_1 + x_2 \leq 30; \\ x_j \geq 0, j = \overline{1,2}. \end{cases}$$
- 21 $f = 2(x_1 - 2)^2 + 3(x_2 - 2)^2 \rightarrow extr$ 22 $z(x, y) = 2x^2 + 2y^2 - 16x + 6y + 1$

$$x_1 + x_2 \leq 12$$

$$\rightarrow extr$$

$$\begin{cases} x + 8y \leq 16; \\ 6x - y = 3. \end{cases}$$
- 23 $f = (x_1 - 1)^2 + (x_2 + 4)^2 \rightarrow extr$ 24 $f = (x_1 - 1)^2 + (x_2 - 3)^2 \rightarrow extr$

$$\begin{cases} 3x_1 + 2x_2 \geq 12 \\ 8x_1 - x_2 \leq 10 \\ -x_1 + 3x_2 \leq 3 \\ x_1 \geq 0 \end{cases}$$

$$\begin{cases} 2x_1 + x_2 \leq 18 \\ -2x_1 + 3x_2 \leq 6 \\ x_1 x_2 \geq 4 \\ x_1 \geq 0, x_2 \geq 0 \end{cases}$$
- 25 $x_1^2 + x_2^2 \rightarrow extr$ 26 $3x_1^2 + 2x_2^2 - 3x_1 + 1 \rightarrow extr$

$$x_1 + x_2 = 1$$

$$x_1^2 + x_2^2 = 4$$

$$27 \quad \begin{aligned} x_1^2 - x_2^2 &\rightarrow \text{extr} \\ x_1 - x_2 &= 4, \\ x_1^2 + x_2^2 &\leq 16 \end{aligned}$$

$$28 \quad \begin{aligned} x + y &\rightarrow \text{extr} \\ (x_1 - 4)^2 + (x_2 - 3)^2 &= 4 \\ 0 \leq x_1 \leq 6, \quad 0 \leq x_2 \leq 4, \end{aligned}$$

$$29 \quad f = (x_1 - 4)^2 + (x_2 - 4)^2 \rightarrow \text{extr} \\ \begin{cases} 3x_1 + 4x_2 \leq 25 \\ x_1 x_2 \geq 5 \\ x_1 \geq 0, x_2 \geq 0 \end{cases}$$

$$30 \quad f = (x_1 - 1)^2 + (x_2 - 1)^2 \rightarrow \text{extr} \\ \begin{cases} 2x_1 + x_2 \leq 10 \\ -2x_1 + 3x_2 \leq 6 \\ x_1 + 2x_2 \geq 4 \end{cases}$$

Додаткові завдання

$$1. F = (x_1 - 4)^2 + (x_2 + 5)^2 \rightarrow \text{max}; \quad 2. F = x_1^2 - 4x_1 + 3x_2 \rightarrow \text{max};$$

$$\begin{cases} x_1 + 2x_2 \leq 14, \\ -5x_1 + 3x_2 \leq 15, \\ 4x_1 + 6x_2 \geq 24, \\ x_1, x_2 \geq 0. \end{cases}$$

$$\begin{cases} 10x_1 + 3x_2 \geq 30, \\ -x_1 + x_2 \leq 5, \\ x_1 + x_2 \leq 10, \\ x_2 \geq 2, \\ x_1, x_2 \geq 0. \end{cases}$$

$$3. F = 1/x_1 + 2x_2 \rightarrow \text{max};$$

$$\begin{cases} 4x_1 - 2x_2 \leq 12, \\ -x_1 + 3x_2 \leq 6, \\ 2x_1 + 4x_2 \geq 16, \\ x_1, x_2 \geq 0. \end{cases}$$

$$4. F = 3x_1^2 + 4x_2 \rightarrow \text{min};$$

$$\begin{cases} 6x_1 + 6x_2 \geq 36, \\ 4x_1 + 8x_2 \geq 32, \\ 2x_1 \geq 2, \quad x_1, x_2 \geq 0. \end{cases}$$

$$5. F = -2x_1 + 1/x_2 \rightarrow \text{min};$$

$$\begin{cases} 3x_1 - 2x_2 \leq 12, \\ -x_1 + 2x_2 \leq 8, \\ 2x_1 + 3x_2 \geq 6, \\ x_1, x_2 \geq 0. \end{cases}$$

$$6. F = -x_1^2 + x_2 \rightarrow \text{max};$$

$$\begin{cases} x_1 + 2x_2 \leq 10, \\ x_1 + 2x_2 \geq 2, \\ 2x_1 + x_2 \leq 10, \\ x_1, x_2 \geq 0. \end{cases}$$

$$7. F = (x_1 + 2)^2 + (x_2 - 3)^2 \rightarrow \text{min};$$

$$\begin{cases} 2x_1 + 4x_2 \leq 16, \\ -4x_1 + 2x_2 \leq 8, \\ x_1 + 3x_2 \geq 9, \\ x_1, x_2 \geq 0. \end{cases}$$

$$8. F = x_1^2 - 2x_2 \rightarrow \text{min};$$

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

$$9. F = 4(x_1 - 1)^2 + (x_2 - 5)^2 \rightarrow \text{max};$$

$$\begin{cases} 6x_1 + 6x_2 \leq 36, \\ 4x_1 + 2x_2 \leq 20, \\ 4x_1 + 8x_2 \leq 40, \\ x_1, x_2 \geq 0. \end{cases}$$

$$10. F = 10x_1^2 + 4x_2^2 \rightarrow \text{min};$$

$$\begin{cases} 5x_1 + 7x_2 \geq 35, \\ 2x_1 \geq 4, \\ x_2 \geq 1, \\ x_1, x_2 \geq 0. \end{cases}$$

Практичне заняття № 3
Постановка задачі нелінійного програмування. Теорема
Веєрштрасса. Задачі безумовної оптимізації

Мета: вивчення умов оптимальності в задачах безумовної оптимізації та класичного методу розв'язання задачі безумовної оптимізації.

План заняття

1. Постановка задачі нелінійного програмування. Точки глобального і локального мінімуму;
2. Теорема Веєрштрасса;
3. Поняття градієнта, похідної за напрямком, гессіана функції;
4. Необхідні і достатні умови оптимальності в задачах безумовної оптимізації, умовної оптимізації;
5. Класичний метод розв'язання задачі НЛП.

Опорні поняття

1. Постановка задачі нелінійного програмування. Точки глобального і локального мінімуму

Сформулюємо типову постановку екстремальної задачі.

Задано множину $\Omega \subseteq \mathbb{R}^N$ і функцію $f(x)$, яка визначена на Ω , $f: \Omega \rightarrow \mathbb{R}$. Необхідно знайти точку x^* , у якій цільова функція набуває найменшого або найбільшого значення на всій допустимій області Ω .

Отже, задача на мінімум має такий вигляд:

$$f(x) \rightarrow \min_{x \in \Omega} \quad (3.1)$$

Якщо $\Omega \subset \mathbb{R}^N$, її називають *задачею умовної оптимізації*. У випадку $\Omega \equiv \mathbb{R}^N$ мають справу із *задачею безумовної оптимізації*.

Елемент $x^* \in \Omega$ називається точкою глобального (абсолютного) мінімуму функції f на множині Ω або глобальним розв'язком задачі (2.1), якщо

$$f(x^*) \leq f(x) \quad \forall x \in \Omega. \quad (3.2)$$

Вочевидь, задача мінімізації не завжди є розв'язною. Наприклад, у випадку $N = 1$ у функцій

$$f(x) = \begin{cases} x^2, & x > 0, \\ 1, & x \leq 0 \end{cases} \quad \text{або} \quad f(x) = e^{-x^2}$$

немає точок мінімуму (див рис. 3.1).

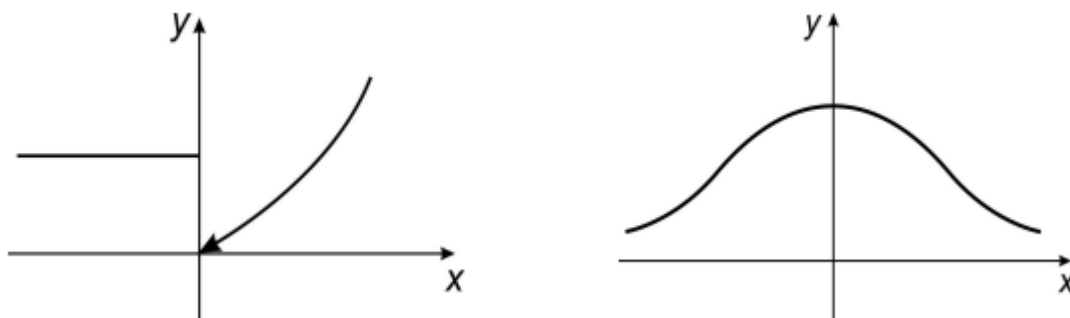


Рис. 3.1 Приклади функцій, які не мають точок мінімуму

У подальшому під час дослідження необхідних умов оптимальності будемо вважати, що задача мінімізації має розв'язок. Проста достатня умова існування розв'язку, яку дає теорема Вейерштрасса, полягає у наступному: якщо за деякого значення $c < \infty$ множина $\{x: f(x) \leq c\}$ є замкненою, непорожньою і обмеженою, а функція $f(x)$ на ній напівнеперервна знизу, то задача мінімізації функції $f(x)$ має розв'язок.

Якщо $\Omega \subset \mathbb{R}^N$, то $\hat{x} \in \Omega$ називається точкою локального (відносного) мінімуму функції f на множині Ω або локальним розв'язком задачі (3.1), якщо існує $\delta > 0$, при якому

$$f(\hat{x}) \leq f(x) \quad \forall x \in \Omega \cap U_\delta(\hat{x}), \quad (3.3)$$

де $U_\delta(\hat{x}) = \{x \in \mathbb{R}^N: \|x - \hat{x}\| \leq \delta\}$ – окіл радіуса $\delta > 0$ точки \hat{x} .

Множину точок глобального мінімуму функції f на Ω позначають $\text{Arg min}_x f(x)$ або Ω^* :

$$\text{Arg min}_x f(x) = \{x \in \Omega \mid f(x) = \min_{x'} f(x')\}.$$

Залежно від властивостей функції f і допустимої множини Ω , Ω^* може бути порожньою, містити скінченну кількість точок або мати потужність континууму. Приміром, розглянемо таку функцію:

$$f(x) = |x - 1| + |x - 2|,$$

і будемо розв'язувати задачу (3.1), коли: 1) $\Omega = [-1, 1]$; 2) $\Omega = [0, 2]$; 3) $\Omega = (3, 5)$.

На рис. 3.2 показано графік функції і результати розв'язування задачі для кожної допустимої області. Так, для $\Omega = [-1, 1]$ функція досягає мінімуму тільки в одній точці, отже, $\Omega^* = \{1\}$. У випадку, коли $\Omega = [0, 2]$, маємо безліч точок мінімуму і $\Omega^* = [1; 2]$. І, нарешті, для області $\Omega = (3, 5)$ мінімум є недосяжним, $\Omega^* = \emptyset$.



Рис. 3.2 Графік функції $f(x) = |x - 1| + |x - 2|$

Визначення 3.1. Функція $f(x)$ є *унімодальною* на заданому відрізку $[a, b]$ тоді і тільки тоді, коли на цьому інтервалі вона є монотонною по обидві сторони від однієї єдиної точки x^* . Тобто коли для будь-яких точок x_1, x_2 виконуються такі умови:

$$x^* \leq x_1 \leq x_2 \rightarrow f(x^*) \leq f(x_1) \leq f(x_2),$$

$$x^* \geq x_1 \geq x_2 \rightarrow f(x^*) \leq f(x_1) \leq f(x_2).$$

Приклади унімодальних функцій показано на рис. 3.3.

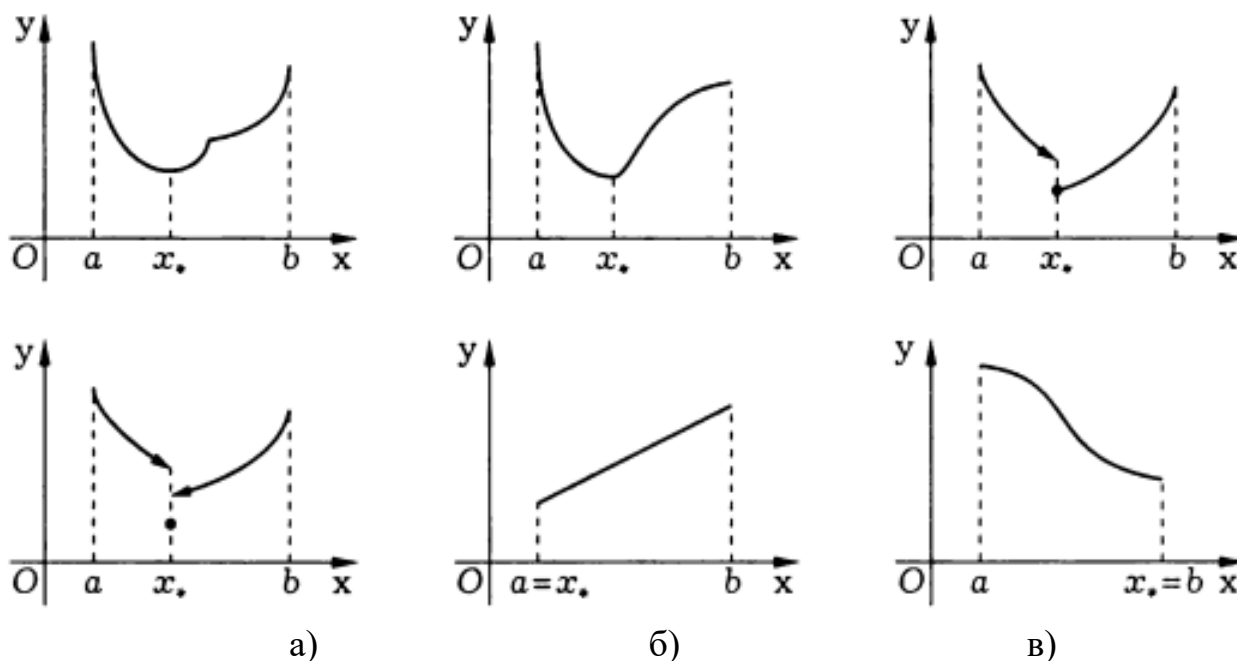


Рис. 3.3. Приклади унімодальних функцій

Для унімодальної функції її локальний мінімум є і глобальним.

Якщо функція не є унімодальною, то можлива наявність кількох локальних оптимумів. При цьому глобальний мінімум можна визначити шляхом знаходження усіх локальних мінімумів і вибору найменшого з них (рис. 3.4).

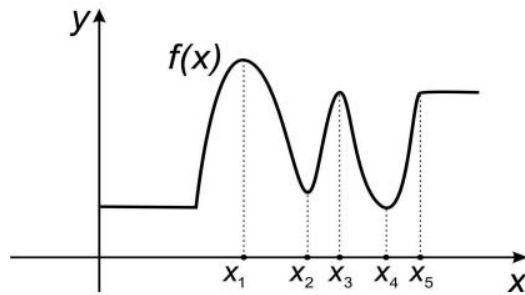


Рис. 4 Приклади екстремумів: x_1 – точка глобального максимуму; x_2, x_4 – точки локальних мінімумів; x_3 – точка локального максимуму; x_5 – точка локального і мінімуму і максимуму

2. Теорема Веєрштрасса

Теорема 3.1. (Веєрштрасса) Нехай Ω – замкнена обмежена підмножина \mathbb{R}^N , $f: \Omega \rightarrow \mathbb{R}$ – неперервна функція. Тоді точка глобального мінімуму f на Ω існує.

У подальшому буде корисною і наступна форма теореми 1.

Теорема 3.2. Нехай Ω – замкнена підмножина \mathbb{R}^N , $f: \Omega \rightarrow \mathbb{R}$ – неперервна функція, до того ж для деякого $x^0 \in \Omega$ множина

$$M(x^0) = \{x \in \Omega \mid f(x) \leq f(x^0)\}$$

є обмеженою. Тоді існує точка мінімуму функції f на множині Ω .

Важливий приклад застосування теореми 2 виникає у зв'язку з наступним поняттям.

Визначення. Проекцією точки $y \in \mathbb{R}^N$ на множину $X \subset \mathbb{R}^N$ називається точка, яка є найближчою до y серед усіх точок множини X , тобто глобальний розв'язок такої задачі

$$|x - y| \rightarrow \min, x \in X.$$

Твердження 1. Проекція довільної точки на будь-яку замкнену множину в \mathbb{R}^N існує.

Твердження 2. Якщо X – опукла підмножина \mathbb{R}^N , то для довільної точки $y \in \mathbb{R}^N$ існує не більше однієї проєкції на X .

Твердження 3. Якщо X – опукла замкнена підмножина \mathbb{R}^N , то для довільної точки $y \in \mathbb{R}^N$ існує одна єдина проєкція на X .

3. Поняття градієнта, похідної за напрямком, гессіана функції

Нехай U – відкрита підмножина \mathbb{R}^N ; $f: U \rightarrow \mathbb{R}^1$ – числова функція на U , $a \in U$.

Скалярна функція $f(x)$ N -вимірному аргументу x називається диференційовною в точці a , якщо існує вектор $p = (p_1, \dots, p_N)^T$, при якому для всіх $h \in \mathbb{R}^N$ буде справедливою рівність

$$f(a + h) - f(a) = (p, h) + o(|h|),$$

де $o(|h|)$ – величина нескінченно мала у порівнянні з $|h|$.

Вектор p , визначений таким чином, називається *градієнтом* функції $f(x)$ в точці a і позначається $f'(a) = \nabla f(a) = \text{grad}f(a)$.

Векторний диференціальний оператор першого порядку

$$\nabla = \frac{\partial}{\partial x_1} \bar{e}_1 + \frac{\partial}{\partial x_2} \bar{e}_2 + \dots + \frac{\partial}{\partial x_N} \bar{e}_N$$

з проєкціями $\frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2}, \dots, \frac{\partial}{\partial x_N}$ на осі координат носить назву *оператор набла* або *оператор Гамільтона*².

Компоненти градієнта називаються *частинними похідними* функції f в точці a і позначаються $\frac{\partial f}{\partial x_1}(a), \frac{\partial f}{\partial x_2}(a), \dots, \frac{\partial f}{\partial x_N}(a)$.

Нагадаємо, що

$$\frac{\partial f(M_0)}{\partial l} = \lim_{M \rightarrow M_0} \frac{f(M) - f(M_0)}{M - M_0}$$

називається *похідною функції f* за напрямком $l = \overline{M_0 M}$. Ця похідна характеризує швидкість зміни функції в точці M_0 за напрямком l . Звичайні частинні похідні $\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_N}$ також можна розглядати як похідні за координатними напрямками.

Відтак, градієнт $\nabla f(a)$ визначається рівністю

$$f(a + h) = f(a) + (\nabla f(a), h) + o(|h|). \quad (3.4)$$

Інакше можна стверджувати, що функція є диференційовною в точці a , якщо вона допускає лінійну апроксимацію в цій точці, тобто знайдеться така лінійна функція

$$\tilde{f}(h) = f(a) + (\nabla f(a), h),$$

що $|f(a + h) - \tilde{f}(h)| = o(|h|)$.

Ясно, що градієнт визначається однозначно. Обчислити його можна безпосередньо за визначенням або за допомогою координатного запису, або ж за правилами диференціювання складної функції.

Функція $f(x)$ називається диференційовною на множині $U \subset \mathbb{R}^N$, якщо вона є диференційовною в кожній точці U .

Для диференційовності функції f в точці a достатньо існування частинних похідних $\frac{\partial f}{\partial x_i}(x)$ в околі точки a і їх неперервності в a .

² Сер Вільям Роуен Гамільтон (англ. William Rowan Hamilton; 4 серпня 1805 – 2 вересня 1865) – ірландський математик, механік-теоретик, фізик-теоретик, "один з кращих математиків XIX століття". Відомий фундаментальними відкриттями в математиці (кватерніони, основи векторного аналізу, варіаційне обчислення, обґрунтування комплексних чисел), аналітичній механіці (гамільтонова механіка) та оптиці.

Функція f називається *двічі диференційовною* в точці a , якщо всі її частинні похідні $\frac{\partial f}{\partial x_i}(x)$ визначені в околі точки a і диференційовні в цій точці. Для двічі диференційовної в точці a функції існують усі другі похідні $\frac{\partial^2 f}{\partial x_i \partial x_j}(a)$.

Отже, скалярна функція $f(x)$ на \mathbb{R}^N називається *двічі диференційовною* в точці a , якщо вона диференційовна в цій точці і знайдеться симетрична $N \times N$ матриця H , при якій для всіх для всіх $h \in \mathbb{R}^N$:

$$f(a+h) = f(a) + (\nabla f(a), h) + \frac{1}{2}(Hh, h) + o(\|h\|^2). \quad (3.5)$$

Матриця H називається *матрицею других частинних похідних, матрицею Гессе* або *гессіаном* і позначається $f''(x) = \nabla^2 f(x)$:

$$\nabla^2 f(x) = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_N} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_N \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_N \partial x_N} \end{pmatrix}.$$

Інакше кажучи, функція є двічі диференційовною в точці a , якщо вона допускає квадратичну апроксимацію в околі цієї точки, тобто існує така квадратична функція

$$\tilde{f}(h) = f(a) + (\nabla f(a), h) + \frac{1}{2}(\nabla^2 f(a)h, h),$$

при якій $|f(a+h) - \tilde{f}(h)| = o(\|h\|^2)$.

Для будь-якого $h \in \mathbb{R}^N$ справедливі такі рівності:

$$\left. \frac{d}{dt} f(a+th) \right|_{t=0} = (f'(a), h), \quad (3.6)$$

$$\left. \frac{d^2}{dt^2} f(a+th) \right|_{t=0} = (f''(a)h, h). \quad (3.7)$$

4. Необхідні і достатні умови оптимальності в задачах безумовної оптимізації, умовної оптимізації

Нагадаємо спочатку деякі відомості з диференціального числення функції однієї змінної.

Нехай функція $f(x)$ однієї змінної x визначена на відкритому інтервалі (a, b) і n -кратно диференційовна на цьому інтервалі. Якщо x^* – внутрішня точка інтервалу, то за теоремою Тейлора можна записати зміну функції f , коли здійснюється перехід від точки x^* до точки $x^* + \varepsilon$:

$$f(x^* + \varepsilon) - f(x^*) = \varepsilon \cdot \left. \frac{df}{dx} \right|_{x=x^*} + \frac{\varepsilon^2}{2!} \cdot \left. \frac{d^2 f}{dx^2} \right|_{x=x^*} + \cdots + \frac{\varepsilon^n}{n!} \cdot \left. \frac{d^n f}{dx^n} \right|_{x=x^*} + o_{n+1}(\varepsilon),$$

де $o_{n+1}(\varepsilon)$ – сума членів, в яких степінь ε дорівнює і вище за $(n+1)$.

Якщо x^* – локальний мінімум функції f на (a,b) , то за визначенням існує такий ε -окіл точки x^* , що для всіх точок x з цього окілу виконується нерівність

$$f(x^*) \leq f(x).$$

А, отже,

$$\varepsilon \cdot \left. \frac{df}{dx} \right|_{x=x^*} + \frac{\varepsilon^2}{2!} \cdot \left. \frac{d^2f}{dx^2} \right|_{x=x^*} + \dots + \frac{\varepsilon^n}{n!} \cdot \left. \frac{d^n f}{dx^n} \right|_{x=x^*} + o_{n+1}(\varepsilon) \geq 0.$$

Коли число ε є достатньо малим, перший доданок домінує над рештою, і за умови можливості вибору ε як додатним, так і від'ємним, остання нерівність буде виконуватися тільки у тому випадку, якщо

$$\left. \frac{df}{dx} \right|_{x=x^*} = 0.$$

Аналогічно міркуючи, неважко помітити, що нерівність буде справедливою тільки тоді, коли

$$\left. \frac{d^2f}{dx^2} \right|_{x=x^*} \geq 0.$$

Така схема застосовна у випадку локального мінімуму. Відтак, отримано результат, який можна сформулювати у вигляді наступної теореми.

Теорема 3.3. Необхідні умови того, що x^* є точкою локального мінімуму (максимуму) дважди диференційовної функції на відкритому інтервалі (a,b) , описуються наступними співвідношеннями:

$$1) \left. \frac{df}{dx} \right|_{x=x^*} = 0, \quad 2) \left. \frac{d^2f}{dx^2} \right|_{x=x^*} \geq 0 (\leq 0).$$

Зауважимо, що ці умови є лише необхідними, тобто коли вони не виконуються, x^* не може бути точкою локального мінімуму (максимуму). Коли ж ці умови виконуються, то немає гарантії того, що x^* – точка локального мінімуму (максимуму). Наприклад, для функції $f = x^3$ точка $x^* = 0$ не є точкою локального мінімуму чи максимуму, хоча $f'(0) = f''(0) = 0$.

Умови екстремуму гладких функцій на всьому просторі добре відомі. Однак їх потрібно розглядати, оскільки вони слугують моделлю, яка дозволяє побудувати аналогічні умови в більш складних випадках.

Розглянемо задачу безумовної оптимізації (задачу без обмежень):

$$f(x) \rightarrow \min, x \in \mathbb{R}^N. \quad (3.8)$$

Теорема 3.4. (Ферма) Якщо \hat{x} – локальний розв'язок задачі (3.8), і функція f диференційовна в точці \hat{x} , то

$$f'(\hat{x}) = 0. \quad (3.9)$$

Точка \hat{x} , яка задовольняє (3.9), називається *стаціонарною* або *критичною* точкою функції f (див. рис. 3.5).

Ця теорема також є лише необхідною умовою. Отже, якщо деяка \hat{x} стаціонарна, вона не обов'язково має бути точкою мінімуму. Вона може бути точкою максимуму або сідловою точкою (точка перегину). Тільки для опуклих функцій така ситуація неможлива, а саме: якщо f – опукла функція, диференційовна в точці \hat{x} , і $\nabla f(\hat{x}) = 0$, то \hat{x} – точка глобального мінімуму f на \mathbb{R}^N .

Відтак, для опуклих функцій необхідна умова екстремуму є і достатньою, причому ця ситуація є загальною і для інших типів опуклих екстремальних задач.

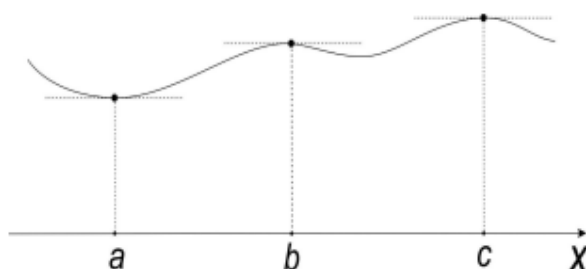


Рис. 3.5 Точки, що задовольняють умову теореми Ферма

Теорема 3.5. (необхідна умова другого порядку) Якщо \hat{x} – локальний розв'язок задачі (3.8), і функція f двічі диференційовна в точці \hat{x} , то

$$(f''(\hat{x})h, h) \geq 0, \quad \forall h \in \mathbb{R}^N, \quad (3.10)$$

(або $f''(\hat{x}) \geq 0$, тобто є невід'ємно визначеною квадратичною формою).

Теорема 3.6. (достатня умова другого порядку) Нехай функція f двічі диференційовна в точці $\hat{x} \in \mathbb{R}^N$, $f'(\hat{x})=0$, а матриця $f''(\hat{x})$ додатно визначена, тобто $(f''(\hat{x})h, h) > 0, \forall h \in \mathbb{R}^N, h \neq 0$. Тоді \hat{x} – локальний розв'язок задачі (3.8).

З курсу вищої алгебри відомо, що існує декілька простих способів, які дозволяють перевірити, чи є дана матриця (а разом з нею і відповідна квадратична форма) додатно визначеною, від'ємно визначеною, невизначеною і т.п. Всі способи перевірки застосовні, лише коли матриця є симетричною.

Справедливі наступні твердження (критерій Сильвестра):

1. Для того, щоб матриця була додатно визначеною ($Q > 0$), необхідно і достатньо, щоб її *послідовні головні мінори* були додатними, тобто

$$D_1 > 0, D_2 > 0, \dots, D_N > 0,$$

$$D_1 = q_{11}, \quad D_2 = \begin{vmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{vmatrix}, \dots, \quad D_N = \begin{vmatrix} q_{11} & \dots & q_{1N} \\ \dots & \dots & \dots \\ q_{N1} & \dots & q_{NN} \end{vmatrix}.$$

2. Аби матриця була Q була невід'ємно визначеною ($Q \geq 0$), необхідно і достатньо, щоб всі її *головні мінори* були невід'ємними:

$$\begin{vmatrix} q_{i_1 i_1} & \dots & q_{i_1 i_p} \\ \dots & \dots & \dots \\ q_{i_p i_1} & \dots & q_{i_p i_p} \end{vmatrix} \geq 0, \quad 1 \leq i_1 \leq i_2 \leq \dots \leq i_p \leq N \quad (p = 1, \dots, N).$$

3. Щоб встановити, що дана матриця є від'ємно визначеною (знаковід'ємною), слід помножити її на (-1) і перевірити отриману матрицю на додатну визначеність (знакододатність).

Або можна скористатися такими критеріями:

1) критерієм від'ємності матриці:

$$(-1)^p \cdot D_p > 0, \quad p = 1, \dots, N;$$

2) критерієм недодатності матриці:

$$(-1)^p \cdot \begin{vmatrix} q_{i_1 i_1} & \dots & q_{i_1 i_p} \\ \dots & \dots & \dots \\ q_{i_p i_1} & \dots & q_{i_p i_p} \end{vmatrix} \geq 0, \quad p = 1, \dots, N.$$

Можна визначати *опуклість* функцій в такий спосіб: якщо матриця Гессе функції додатно визначена, то функція опукла.

Примітка. під час перевірки матриці на невизначеність достатньо переконатися в тому, що принаймні два діагональні елементи мають різні знаки, приміром:

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

На рис. 3.6 зображено лінії рівня деякої функції, яка має дві точки локального мінімуму і сідлову точку.

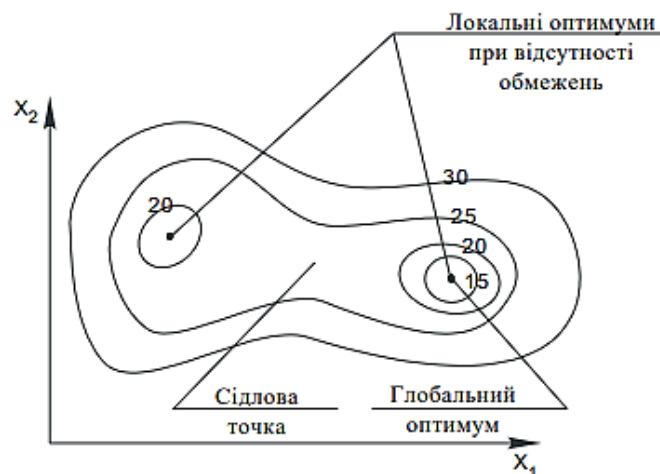


Рис. 3.6. Приклад функції двох змінних з декількома локальними оптимумами

5. Класичний метод розв'язання задачі безумовної оптимізації

Опишемо алгоритм класичного методу розв'язання задачі (3.6).

1. Виписати необхідні умови оптимальності (3.9).
2. Знайти всі стаціонарні точки функції. Для цього розв'язати систему рівнянь (3.9).
3. Серед критичних (стаціонарних) точок визначити точки мінімуму цільової функції, використовуючи достатні умови оптимальності другого порядку.

Приклад 3.1. Для поданої нижче функції

$$f(x) = 2x_1^2 + 4x_1x_2^3 - 10x_1x_2 + x_2^2$$

визначити класифікацію точки $\hat{x} = (0,0)$.

Розв'язання. Знайдемо градієнт функції, а саме:

$$\frac{\partial f}{\partial x_1} = 4x_1 + 4x_2^3 - 10x_2, \quad \frac{\partial f}{\partial x_2} = 12x_1x_2^2 - 10x_1 + 2x_2,$$

$$\nabla f(\hat{x}) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

Отже, \hat{x} – стаціонарна точка.

Обчислюємо гессіан функції $f(x)$, тобто

$$\frac{\partial^2 f(\hat{x})}{\partial x_1^2} = 4, \quad \frac{\partial^2 f(\hat{x})}{\partial x_1 \partial x_2} = 12x_2^2 - 10|_{x=(0,0)} = -10,$$

$$\frac{\partial^2 f(\hat{x})}{\partial x_2^2} = 24x_1x_2 + 2|_{x=(0,0)} = 2,$$

$$H = \nabla^2 f(\hat{x}) = \begin{pmatrix} 4 & -10 \\ -10 & 2 \end{pmatrix}.$$

Матриця $\nabla^2 f(\hat{x})$ є невизначеною, оскільки за критерієм Сільвестра $D_1 > 0$, $D_2 < 0$.

Тому $\hat{x} = (0,0)$ є сідловою точкою функції $f(x)$ (див. рис. 3.7).

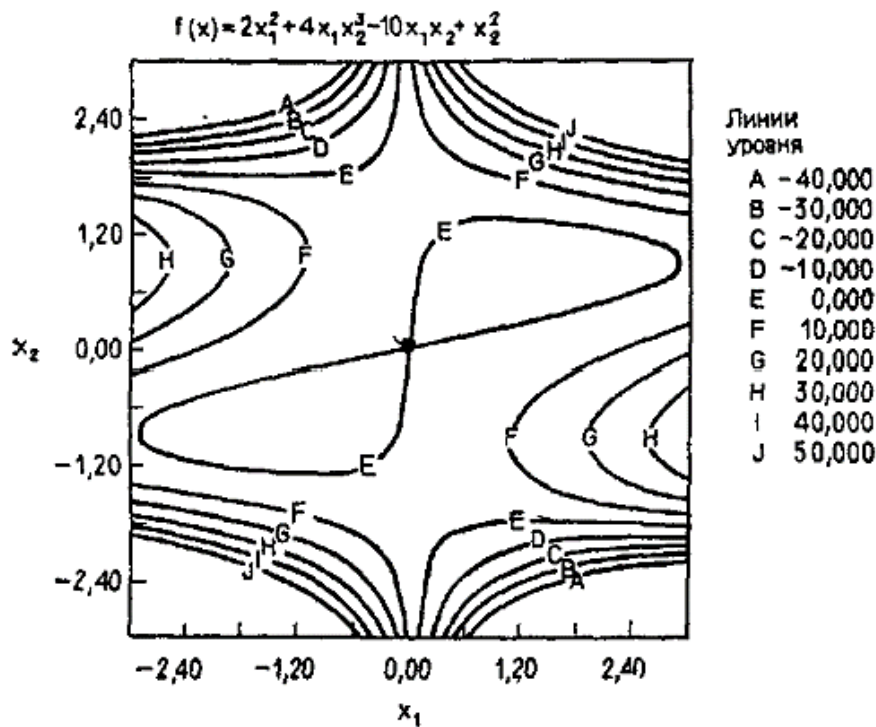


Рис. 3.7 Лінії рівня досліджуваної функції

Приклад 3.2. Побудувати поверхню функції Хіммельблау:

$$f(x) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2.$$

Розв'язання. Наведемо програмний код на Python, який будує поверхню функції в діапазоні $[-5,5] \times [-5,5]$.

```
# multimodal test function
from numpy import arange
from numpy import meshgrid
from matplotlib import pyplot
from mpl_toolkits.mplot3d import Axes3D
# objective function
def objective(x, y):
    return (x**2 + y - 11)**2 + (x + y**2 - 7)**2
# define range for input
r_min, r_max = -5.0, 5.0
# sample input range uniformly at 0.1 increments
xaxis = arange(r_min, r_max, 0.1)
yaxis = arange(r_min, r_max, 0.1)
# create a mesh from the axis
x, y = meshgrid(xaxis, yaxis)
# compute targets
results = objective(x, y)
# create a surface plot with the jet color scheme
figure = pyplot.figure()
axis = figure.gca(projection='3d')
axis.plot_surface(x, y, results, cmap='jet')
# show the plot
pyplot.show()
```

Результатом роботи програми є поверхня, зображена на рис. 3.8.

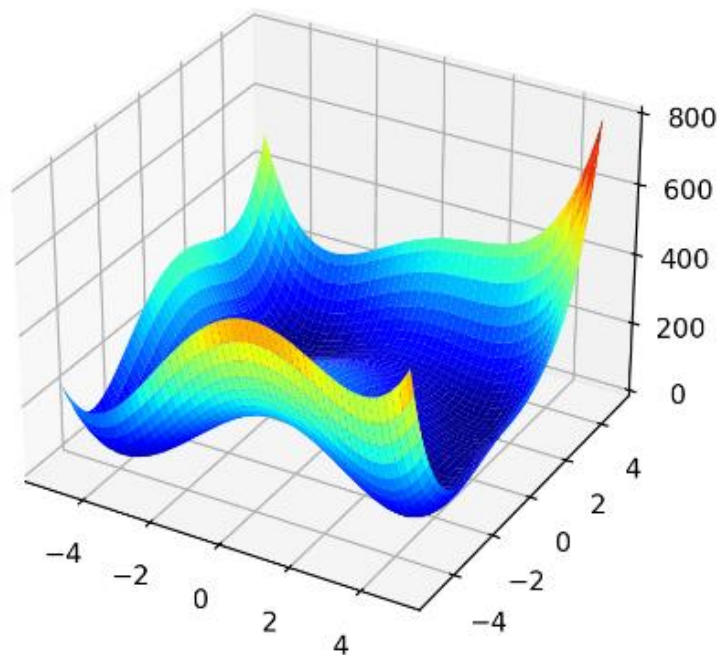


Рис. 3.8 Графічне зображення функції Хіммельблау

Функція має чотири точки мінімуму: $(3,0; 2,0)$, $(-2,805118; 3,131312)$, $(-3,779310; -3,283186)$, $(3,584428; -1,848126)$.

Приклад 3.3. Для функції $f(x) = x_1^3 + x_2^3 - 3x_1x_2$ знайти стаціонарні точки і визначити їх класифікацію.

Розв'язання. Запишемо градієнт функції:

$$\frac{\partial f}{\partial x_1} = 3x_1^2 - 3x_2, \quad \frac{\partial f}{\partial x_2} = 3x_2^2 - 3x_1$$

і розв'яжемо отриману систему рівнянь

$$\nabla f(\hat{x}) = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \Leftrightarrow \begin{cases} 3x_1^2 - 3x_2 = 0, \\ 3x_2^2 - 3x_1 = 0. \end{cases} \Leftrightarrow \begin{cases} x_2 = x_1^2, \\ (x_1^3 - 1)x_1 = 0. \end{cases}$$

Як бачимо, система має два розв'язки: $\hat{x}^{(1)} = (0,0)$; $\hat{x}^{(2)} = (1,1)$ – це стаціонарні точки функції.

Гессіан функції $f(x)$ має такий вигляд:

$$H = \nabla^2 f(x) = \begin{pmatrix} 6x_1 & -3 \\ -3 & 6x_2 \end{pmatrix}.$$

В стаціонарній точці $\hat{x}^{(1)}$ матриця H є невизначеною, отже, робимо висновок, що $\hat{x}^{(1)}$ не є точкою максимуму або мінімуму. З додатної визначеності гессіана в точці $\hat{x}^{(2)} = (1,1)$ випливає, що остання є точкою локального мінімуму.

Приклад 3.4. Дослідити на екстремум таку функцію: $f(x) = (x^2 - 1)^3$.

Розв'язання. Градієнтом скалярної функції є її похідна, а саме:

$$f'(x) = 3(x^2 - 1)^2 \cdot 2x = 6x(x - 1)^2(x + 1)^2.$$

$f'(x) = 0$, якщо $x = 0, x = 1, x = -1$. В цих критичних точках виконуються необхідні умови оптимальності першого порядку. Далі знаходимо другу похідну і визначаємо її знак в знайдених точках:

$$f''(x) = 6(x^2 - 1)(5x^2 - 1) = H(x).$$

$$H(1) = H(-1) = 0, \quad H(0) = 6.$$

Відтак, матриця Гессе додатньо напіввизначена в усіх трьох точках, тобто виконуються необхідні умови мінімуму другого порядку. Але лише для точки $x = 0$ маємо $H(0) > 0$, тобто критична точка $x = 0$ є точкою глобального мінімуму.

Приклад 3.5. Квадратична функція

$$f(x_1, x_2) = 6x_1^2 - 4x_1x_2 + 3x_2^2 + 4\sqrt{5}(x_1 + 2x_2) + 22 \quad (3.11)$$

буде далі використовуватися під час порівняння різних обчислювальних методів безумовної оптимізації, виявлення їх недоліків і переваг. Тому розглянемо детально її властивості.

Зауважимо, що функцію $f(x_1, x_2)$ можна подати у такому вигляді:

$$f(x) = (Ax, x) + (b, x) + c, \quad (3.12)$$

де $x = (x_1, x_2) \in R^2$, $A = \begin{pmatrix} 6 & -2 \\ -2 & 3 \end{pmatrix}$, $b = \begin{pmatrix} 4\sqrt{5} \\ 8\sqrt{5} \end{pmatrix}$, $c = 22$.

Матриця A є додатно визначеною, оскільки має такі кутові мінори:

$\Delta_1 = 6 > 0$, $\Delta_2 = 6 \cdot 3 - 4 = 14 > 0$. Відтак, функція є сильно опуклою у просторі R^2 і має одну єдину стаціонарну точку x^* , яка є точкою її мінімуму. Цю точку можна знайти за формулою $x^* = -H^{-1}b$, де $H = 2A$ є матрицею Гессе даної функції.

З того, що

$$A^{-1} = \frac{1}{14} \begin{pmatrix} 3 & 2 \\ 2 & 6 \end{pmatrix},$$

маємо

$$x^* = -\frac{1}{28} \begin{pmatrix} 3 & 2 \\ 2 & 6 \end{pmatrix} \begin{pmatrix} 4\sqrt{5} \\ 8\sqrt{5} \end{pmatrix} = \begin{pmatrix} -\sqrt{5} \\ -2\sqrt{5} \end{pmatrix}.$$

В цій точці $f(x^*) = -28$.

Зведемо квадратичну форму (Ax, x) функції до канонічного вигляду. Для цього знайдемо власні значення матриці A , склавши її характеристичне рівняння $\det(A - \lambda I) = 0$, де I – одинична матриця другого порядку.

У даному випадку це рівняння має такий вигляд:

$$\begin{vmatrix} 6 - \lambda & -2 \\ -2 & 3 - \lambda \end{vmatrix} = 0.$$

Розкриваючи визначник ліворуч, отримаємо, що $(6 - \lambda)(3 - \lambda) - 4 = 0$. Корені цього рівняння $\lambda = 2$ і $\lambda = 7$, які і є власними значеннями матриці A . Оскільки власні значення різні, а матриця A симетрична, відповідні цим власним значенням власні вектори є ортогональними.

Координати власних векторів знайдемо, розв'язуючи систему лінійних алгебраїчних рівнянь $(A - \lambda I)e = 0$, яка за умов $\lambda = 2$ і $\lambda = 7$ зводиться до таких систем:

$$\begin{cases} 4x_1 - 2x_2 = 0, \\ -2x_1 + x_2 = 0, \end{cases} \quad \begin{cases} -x_1 - 2x_2 = 0, \\ -2x_1 - 4x_2 = 0. \end{cases}$$

Розв'язком першої з них є вектор $\begin{pmatrix} 1 \\ 2 \end{pmatrix}$, а другої – $\begin{pmatrix} -2 \\ 1 \end{pmatrix}$. Нормуючи ці вектори, отримуємо одиничні вектори

$$e_1 = \begin{pmatrix} 1/\sqrt{5} \\ 2/\sqrt{5} \end{pmatrix} \text{ і } e_2 = \begin{pmatrix} -2/\sqrt{5} \\ 1/\sqrt{5} \end{pmatrix},$$

які утворюють в R^2 ортонормований базис. У цьому базисі матрицею квадратичної форми є діагональна матриця Λ з діагональними елементами, котрі дорівнюють власним значенням, а квадратична форма (Ax, x) набуває канонічного вигляду $(Ay, y) = \lambda_1 y_1^2 + \lambda_2 y_2^2 = 2y_1^2 + 7y_2^2, y = (y_1, y_2)$.

Координати власних векторів e_1 і e_2 , записані по стовпчиках, утворюють матрицю переходу U до нового, канонічного базису.

В даному випадку матриця переходу має такий вигляд:

$$U = \begin{pmatrix} 1/\sqrt{5} & -2/\sqrt{5} \\ 2/\sqrt{5} & 1/\sqrt{5} \end{pmatrix} = \frac{1}{\sqrt{5}} \begin{pmatrix} 1 & -2 \\ 2 & 1 \end{pmatrix}.$$

Оскільки матриця U визначає перехід від одного ортонормованого базису до іншого, вона є ортогональною. Тому обернену матрицю U^{-1} можна знайти за допомогою транспонування, тобто $U^{-1} = U^T$. Використовуючи матрицю переходу можна зйти координати вектора b у новому базисі за формулою $b^0 = U^{-1}b$, тобто

$$b^0 = \frac{1}{\sqrt{5}} \begin{pmatrix} 1 & -2 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} 4/\sqrt{5} \\ 8/\sqrt{5} \end{pmatrix} = \begin{pmatrix} 20 \\ 0 \end{pmatrix}.$$

Отже, квадратична функція в новій системі координат приймає вигляд

$$f_1(y_1, y_2) = 2y_1^2 + 7y_2^2 + 20y_1 + 22.$$

За допомогою виділення повних квадратів проводимо подальше спрощення вигляду функції:

$$f_1(y_1, y_2) = 2(y_1 + 5)^2 + 7y_2^2 - 28.$$

або

$$f_2(z_1, z_2) = 2z_1^2 + 7z_2^2 - 28, \text{ де } z_1 = y_1 + 5 \text{ і } z_2 = y_2.$$

З отриманого подання функції випливає, що її графіком є еліптичний параболоїд.

Зауважимо, що перетворення функції здійснено у два кроки: спочатку перехід до змінних y_1, y_2 , а далі – до змінних z_1, z_2 . Формули зв'язку вихідних змінних x_1, x_2 з кінцевими змінними z_1 і z_2 мають такий вигляд:

$$\begin{cases} x_1 = \frac{z_1 - 2z_2}{\sqrt{5}} - \sqrt{5}, \\ x_2 = \frac{2z_1 + z_2}{\sqrt{5}} - 2\sqrt{5}. \end{cases}$$

Початком остаточної системи координат є точка O , яка в початковій системі координат має координати $x_1 = -\sqrt{5}$, $x_2 = -2\sqrt{5}$. Всі три системи координат зображено на рис. 3.9.

Лініями рівня функції $f(x_1, x_2)$ є подібні еліпси (рис. 3.9). Знайдемо, приміром, рівняння лінії рівня, що проходить через точку $x^0 = (-2, 1)$. Підстановка координат цієї точки в рівняння (3.11) дає значення функції $f(x^0) = 57$. Відтак, ця точка знаходиться на лінії рівня $f(x_1, x_2) = 57$. Переходячи до остаточної системи координат $O_1 z_1 z_2$, отримаємо рівняння $f_2(z_1, z_2) = 2z_1^2 + 7z_2^2 - 28 = 57$, або

$$\frac{z_1^2}{\frac{85}{2}} + \frac{z_2^2}{\frac{85}{7}} = 1.$$

Відтак, шукана лінія рівня – еліпс з півосями $a = \sqrt{85/2}$ і $b = \sqrt{85/7}$, який виділений на рис. 3.9 товстою лінією.

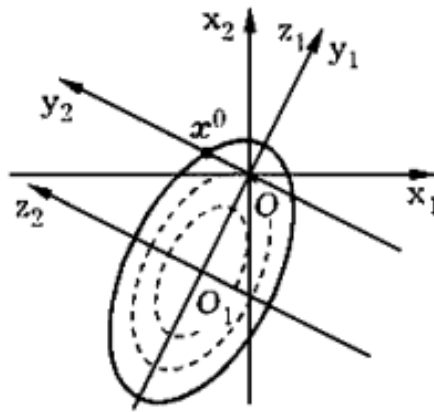


Рис. 3.9. Ілюстрація до прикладу 5

Рекомендована література та інформаційні ресурси

1. Ржевський С.В. Дослідження операцій / С.В. Ржевський, В.М.Александрова. – К.: Академвидав, 2006. – 560 с.
2. Математичні методи дослідження операцій: підручник / Є.А. Лавров, Л.П. Пертун, В.В. Шендрік та ін. – Суми : Сумський державний університет, 2017. – 212 с.
3. Нефьодов Ю.М. Методи оптимізації в прикладах і задачах : навчальний посібник / Ю.М. Нефьодов, Т.Ю. Балицька. – Київ : Кондор, 2011. – 324 с.
4. Бех О.В. Математичне програмування : навч. посібн. / О.В. Бех, Т.А. Городня, А.Ф. Щербак. – Львів : Магнолія–2006, 2014. – 200 с.
5. Методи оптимізації : Методичні вказівки до виконання лабораторних робіт для студентів напряму підготовки 6.050101 «Комп'ютерні науки», спеціальностей 8.05010102 «Інформаційні технології проектування» та 8.05010103 «Системне проектування» денної та заочної форм навчання / Укл. В.В. Бобін, В.В. Ладогубець, О.Д. Фіногенов. – К. : НТУУ «КПІ», 2011 р. – 128 с.
6. Оптимізація систем керування.: навч. посіб. для студ. спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» / Л. Р. Ладієва; КПІ ім. Ігоря Сікорського. 2020. – 211с.

Завдання для розв'язування у класі

1. Використовуючи геометричні побудови, з'ясувати, чи мають подані нижче задачі глобальний розв'язок.

а) $-x_1 + x_2 \rightarrow \min, x \in D,$

$$D = \{x \in \mathbb{R}^2 \mid x_1^2 + x_2^2 \leq 1, x_1 \geq x_2^2, x_1 + x_2 \geq 0\};$$

б) $\max \{|x_1 - 2|, |x_2|\} \rightarrow \min, x \in D,$

$$D = \{x \in \mathbb{R}^2 \mid 2|x_1| - |x_2| \leq 2\};$$

в) $(x_1 - 1)^2 + (x_2)^2 \rightarrow \min, x \in D,$

$$D = \{x \in \mathbb{R}^2 \mid x_1^2 + x_2^2 = 2\sigma x_1 x_2\}, \sigma \in \mathbb{R} - \text{параметр.}$$

2. Використовуючи геометричні побудови, висунути гіпотезу, за яких значень параметра σ задача оптимізації:

$$x_1 + \sigma x_2 \rightarrow \min, x \in D,$$
$$D = \{x \in \mathbb{R}^2 \mid x_1^3 + x_2^3 = 3x_1x_2\}$$

має глобальний розв'язок, а за яких – лише локальний.

3. Визначити та класифікувати стаціонарні точки таких функцій:

а) $f(x) = x_1^2 - x_1x_2 + x_2^2 - 6x_1 + 8x_2 - 7,$

б) $f(x) = 2x_1^3 + 4x_1x_2^2 - 10x_1x_2 + x_2^2 + 12,$

в) $f(x) = x_1^2 - 6x_1x_2 + 2x_2^2 + 3x_3^2 + 8x_1x_3 - 4x_2x_3 - 5x_1 + 3,$

г) $f(x) = x_1^2 + 4x_1x_2 + 5x_2^2 + 3x_3^2 - 2x_1x_3 - 2x_2x_3.$

4. За яких значень параметрів σ і μ функція буде опуклою?

а) $f(x) = x_1^2 - 2\sigma x_1x_2 + \mu x_2^2 - 16x_1 + 2x_2 - 27,$

б) $f(x) = 2\mu x_1^2 + 4x_2^2 - 10\sigma x_1x_2 + 12,$

в) $f(x) = 5x_1^2 - \sigma x_1x_2 + 2x_2^2 + 3x_3^2 + \mu x_1x_3 - x_2x_3 + 3,$

г) $f(x) = \sigma x_1^2 + 10x_1x_2 + x_2^2 + \mu x_3^2 - 2x_1x_3 - 2\mu x_2x_3.$

Контрольні запитання

1. Дайте визначення глобального (локального) розв'язку задачі оптимізації.

2. Сформулюйте критерій оптимальності першого порядку для задачі нелінійної оптимізації.

3. Яка умова є критерієм оптимальності другого порядку в задачах оптимізації?

4. Дайте визначення градієнта цільової функції, опишіть його властивості.

5. Що являє собою гессіан функції?

Завдання для самостійної та індивідуальної роботи

Відповідно до варіанта виконати наступні завдання:

1) Записати градієнт функції $f(x)$ в заданій точці M і знайти похідну функції $f(x)$ за напрямком вектору \bar{a} ; зростає чи спадає функція за даним напрямком?

2) Записати матрицю Гессе функції $f(x)$ і визначити її компоненти в заданій точці M .

Варіанти завдань

№ варіанта	Функція	Точка M	Вектор \bar{a}
1	$4x_1^2 - 3x_1x_2 + x_2^3 - 6x_1 + 8x_2 - 27$	(2,3)	(-3,4)
2	$f(x) = x_1^2 - x_1x_2 + 6x_1 - x_2 - 2$	(4,-1)	(5,12)
3	$f(x) = 5x_1^3 - x_1x_2 + x_1x_2^2 - 6x_1 - 1$	(3,-3)	(-1,1)
4	$f(x) = 2x_1^2 - x_1x_2 + x_2^2 - 8x_1 + 8x_2 - 7$	(1,1)	(7,15)
5	$f(x) = 4x_1^2 - 7x_1x_2 + x_2^2 - 12x_1 + 8x_2 + 6$	(-3,5)	(-3,-4)
6	$f(x) = 5x_1^2 - x_1x_2 + x_2^2 - 23x_1 + 8x_2 - 5$	(2,2)	(5,12)
7	$f(x) = 8x_1^2 - 3x_1x_2 + x_2^2 - 6x_1 + 8x_2$	(5,-2)	(-1,1)
8	$f(x) = 5x_1^2 - 6x_1x_2 + x_2^2 - 6x_1 + 8x_2$	(4,1)	(7,15)
9	$f(x) = 9x_1^2 + 3x_1x_2 + x_2^3 - 6x_1 + 8x_2$	(-2,-3)	(-3,-4)
10	$f(x) = 3x_1^2 - 5x_2^2 - 6x_1 + 8x_2$	(2,3)	(5,12)
11	$f(x) = -4x_1^2 - 11x_1x_2 + x_2^2 - 42x_1 + x_2$	(4,-1)	(-1,-1)
12	$f(x) = -2x_1^2 - x_2^4 - 6/x_1 + 8x_2 - 7$	(3,-3)	(7,-15)
13	$f(x) = -3x_1^2 - 13x_1x_2 + x_2^2 - 6x_1 + 8/x_2$	(1,1)	(-3,4)
14	$f(x) = -8x_1^2 - x_1/x_2 + x_2^2 - 6x_1 + 8x_2 - 7$	(-3,5)	(5,12)
15	$f(x) = 5x_1^2 - 17x_1x_2 + x_2^2 - 6x_1 + 8x_2 - 7$	(2,2)	(-1,1)
16	$f(x) = -4x_1^2 - 16x_1x_2 + x_2^2 - 2/x_1 + 8x_2$	(5,-2)	(-7,15)
17	$f(x) = 10x_1^2 - 17x_1x_2 + x_2^3 - 6x_1 + 8x_2 - 7$	(4,1)	(-3,-4)
18	$f(x) = 9x_1^2 - 18x_1x_2 + x_2^2 - 6x_1 + 8x_2 - 1$	(-2,-3)	(5,12)
19	$f(x) = -4x_1^2 - x_1/x_2 - 3x_2^2 - 6x_1 - x_2 - 7$	(2,3)	(-1,1)

№ варіанта	Функція	Точка М	Вектор \bar{a}
20	$f(x) = 7x_1^2 - x_1x_2 + x_2^2 - 6x_1 - 4x_2 - 2$	(4,-1)	(7,-15)
21	$f(x) = x_1^2 - 8x_1/x_2 + x_2^2 - 6x_1 - 18x_2$	(3,-3)	(-3,4)
22	$f(x) = 4x_1^2 - x_1x_2 + x_2^4 - 5/x_1 + 18x_2$	(1,1)	(5,12)
23	$f(x) = x_1^4 - 23x_1x_2 + x_2^2 - 6x_1 + 33x_2 - 7$	(-3,5)	(-1,1)
24	$f(x) = 8x_1^2 - x_1x_2 + 5x_2^2 + x_1 + 23/x_2$	(2,2)	(-7,15)
25	$f(x) = x_1^2 - x_1x_2 + x_2^2 - 6x_1 + 4x_2$	(5,-2)	(-3,4)
26	$f(x) = -8x_1^2 - x_1x_2 + x_2^2 + 4x_1 + 7/(x_2 - 7)$	(4,1)	(5,-12)
27	$f(x) = -4x_1^2 + 7x_1x_2 + 3x_2^2 - 6/x_1 + 6x_2$	(-2,-3)	(-1,1)
28	$f(x) = x_1^2 - 3x_1x_2 + x_2^2 - x_1 + 8x_2$	(-1,1)	(7,15)
29	$f(x) = -x_1^2 - 2x_2^2 - 6x_1 + 3/(x_2 - 7)$	(-4,3)	(-2,2)
30	$f(x) = 8x_1^2 - 2x_1x_2 + x_2^3 - 5x_1 + 8x_2$	(1,-3)	(6,-8)

Додаткові завдання

Для заданої функції знайти критичні точки і визначити їх тип.

1. $F(x) = 2x^3 - 6x^2 - 18x - 10$;
2. $F(x) = -3x^2 + 4x - 2$;
3. $F(x) = 2(x - 2)^2 - 1$;
4. $F(x) = -x^3 + 3x^2 - 4$;
5. $F(x) = \frac{x}{1+x^2}$;
6. $F(x) = -(x + 2)^2(x - 1)^2$;
7. $F(x) = x + \frac{1}{x}$;
8. $F(x) = (x - 1)^3(x + 1)^2$.

Практичне заняття № 4

Задачі з обмеженнями. Метод множників Лагранжа

Мета: вивчення методу редукції задачі умовної оптимізації до задачі на безумовний екстремум.

План заняття

1. Постановка задачі оптимізації з обмеженнями;
2. Розв'язання задач оптимізації з обмеженнями-рівностями методом виключення змінних;
3. Метод множників Лагранжа.

Опорні поняття

1. Постановка задачі оптимізації з обмеженнями

Раніше були розглянуті необхідні та достатні умови оптимальності розв'язків оптимізаційних задач без обмежень. Однак існує низка інженерних завдань, які пов'язані з оптимізацією за наявності певних умов щодо керованих змінних. І хоча такі обмеження суттєво зменшують розміри області, в якій проводиться пошук оптимуму, процес оптимізації стає більш складним, оскільки встановлені вище критерії оптимальності не можна використовувати при обмеженнях. Навіть основна умова, відповідно до якої оптимум досягається в стаціонарній точці, що характеризується нульовим градієнтом, може порушуватися.

Наприклад, безумовний мінімум функції $f(x) = (x - 2)^2$ має місце в стаціонарній точці $x = 2$. Але якщо задача мінімізації розв'язується з урахуванням обмеження $x \geq 4$, то буде знайдено умовний мінімум, якому відповідає точка $x = 4$ (рис. 4.1). Вона не є стаціонарною точкою функції f , оскільки $f'(4) = 4$. Отже, необхідно сформулювати нові умови оптимальності для розв'язків задач з обмеженнями.

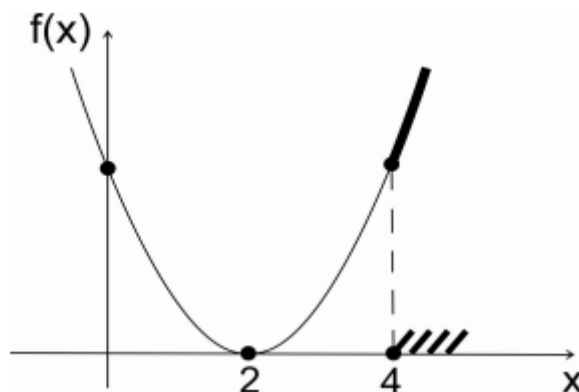


Рис. 4.1 Функція $f(x) = (x - 2)^2$ та її умовний мінімум

Розглянемо окремий випадок загальної задачі нелінійного програмування, система обмежень якої складається лише з рівнянь, відсутні умови невід'ємності змінних і функцій f і g_i неперервні разом зі своїми частинними похідними:

$$f(x) \rightarrow \min (\max), x \in \Omega, \quad (4.1)$$

$$\Omega = \{x \in \mathbb{R}^N: g_i(x) = 0, 1 \leq i \leq s, s < N\}. \quad (4.2)$$

В курсі математичного аналізу задачу (4.1), (4.2) називають *класичною задачею оптимізації* або *задачею на умовний екстремум*.

2. Розв'язання задач оптимізації з обмеженнями-рівностями шляхом зменшення їх розмірності

Задачу (4.1), (4.2) можна звести до безумовної оптимізації шляхом виключення з цільової функції s незалежних змінних за допомогою заданих рівностей, зменшуючи розмірність вихідної задачі з N до $N - s$.

Приклад 4.1. розглянемо таку задачу оптимізації із обмеженнями-рівностями:

$$f(x) = x_1 x_2 x_3 \rightarrow \min,$$

$$g_1(x) = x_1 + x_2 + x_3 - 1 = 0.$$

Розв'язання. З рівності $g_1(x) = 0$ отримаємо вираз для x_3 через решту змінних і виключимо x_3 з цільової функції. Таким чином ми отримуємо оптимізаційну задачу з двома змінними без обмежень, а саме:

$$f(x_1, x_2) = x_1 x_2 (1 - x_1 - x_2) \rightarrow \min.$$

Для її розв'язання можна застосовувати класичний метод оптимізації, заснований на необхідних умовах екстремуму. Отже, знайдемо похідні і прирівняємо їх до нуля, а саме:

$$\frac{\partial f(x_1, x_2)}{\partial x_1} = x_2(1 - x_1 - x_2) - x_1 x_2 = 0,$$

$$\frac{\partial f(x_1, x_2)}{\partial x_2} = x_1(1 - x_1 - x_2) - x_1 x_2 = 0.$$

Розв'язуючи цю систему рівнянь, отримуємо чотири стаціонарних точки функції $f(x_1, x_2)$: $(0;0)$, $(0;1)$, $(1;0)$; $(0,6; -0,2)$. Тепер, використовуючи необхідні і достатні умови оптимальності другого порядку, можна визначити характер кожної з точок.

Умовно стаціонарними для вихідної функції $f(x)$ будуть такі точки: $(0; 0; 1)$, $(0; 1; 0)$, $(1; 0; 0)$; $(0,6; -0,2; 0,6)$.

Однак метод виключення змінних використовують лише в тих випадках, коли систему рівнянь обмежень можна розв'язати відносно деякого конкретного набору незалежних змінних. Часто цей процес стає дуже трудомісткою процедурою, якщо взагалі можливою. У такому разі доцільно використовувати *метод множників Лагранжа*.

3. Метод множників Лагранжа

Метод множників Лагранжа дозволяє встановити необхідні умови, що дозволяють ідентифікувати точки оптимуму в задачі з обмеженнями-рівностями. При цьому вона перетворюється на еквівалентну задачу безумовної оптимізації із деякими невідомими параметрами, які називаються *множниками Лагранжа*.

Введемо до розгляду функцію Лагранжа для задачі (4.1), (4.2), а саме:

$$L(x, \lambda) = \lambda_0 f(x) + \sum_{k=1}^s \lambda_k g_k(x),$$

де параметри $\lambda_0, \dots, \lambda_s$ – множники Лагранжа.

Прирівнюючи частинні похідні функції L по x до нуля, отримаємо систему з N рівнянь відносно $N + s$ невідомих, тобто

$$\begin{cases} \frac{\partial L(x, \lambda)}{\partial x_1} = 0, \\ \dots \\ \frac{\partial L(x, \lambda)}{\partial x_N} = 0. \end{cases}$$

Якщо знайти розв'язок системи у вигляді функцій вектора λ неможливо, то систему розширюють шляхом включення до неї обмежень задачі: $g_i(x) = 0, 1 \leq i \leq s$. Далі проводять процедуру перевірки стаціонарних точок на основі обчислення елементів матриці Гессе.

Частинні похідні функції Лагранжа за координатами вектора x мають такий вигляд:

$$\frac{\partial L(x, \lambda)}{\partial x_j} = \lambda_0 \frac{\partial f(x)}{\partial x_j} + \sum_{k=1}^s \lambda_k \frac{\partial g_k(x)}{\partial x_j}, \quad j = 1, 2, \dots, N.$$

Складений з них вектор можна позначити наступним чином:

$$L'_x(x, \lambda_0, \Lambda) = \lambda_0 \nabla f(x) + \sum_{k=1}^s \lambda_k \nabla g_k(x). \quad (4.3)$$

Справедлива наступна теорема про необхідні умови локальної оптимальності в задачі (4.1), (4.2), відома як *правило множників Лагранжа*.

Теорема 1. (Правило множників Лагранжа) Нехай функції $f(x), g_k(x), 1 \leq k \leq s$ неперервно диференційовні в деякому околі точки

$x^* \in \mathbb{R}^N$. Якщо x^* – локальний розв’язок задачі (4.1), (4.2), то існує число λ_0^* і вектор $\Lambda^* = (\lambda_1^*, \dots, \lambda_s^*)$, котрі не дорівнюють одночасно нулю і при яких

$$L'_x(x^*, \lambda_0^*, \Lambda^*) = 0. \quad (4.4)$$

Якщо градієнти $\nabla f(x), \nabla g_k(x), 1 \leq k \leq s$ є лінійно незалежними (умова регулярності), то $\lambda_0^* \neq 0$.

Умова (4.4) при цьому називається *умовою стаціонарності* функції Лагранжа.

З урахуванням виразу (4.3) умова (4.4) означає, що градієнти $\nabla f(x)$ і $\nabla g_k(x), 1 \leq k \leq s$ є лінійно залежними. Зокрема, якщо $s = 1$, то вони колінеарні.

Будь-яка x^* , що задовольняє за деяких λ_0^* і Λ^* умову (4.4) і умови допустимості $g_i(x) = 0, 1 \leq i \leq s$, називається *стаціонарною точкою* задачі (4.1), (4.2). Вона визначається з системи $N + s$ рівнянь з $N + s + 1$ невідомими. Зауважимо, що у випадку $\lambda_0^* \neq 0$ завжди можна вважати $\lambda_0^* = 1$ (для цього достатньо поділити всі множники Лагранжа на λ_0^*), тобто невідомих в системі буде також $N + s$.

Тому, якщо виконується визначена в теоремі умова регулярності, то можна обмежитися розглядом функції Лагранжа такого вигляду:

$$L(x, \Lambda) = f(x) + \sum_{k=1}^s \lambda_k g_k(x),$$

яку називають *регулярною*.

Як і в задачі безумовної оптимізації, стаціонарні точки не обов’язково є розв’язками (4.1), (4.2). Тут також існують необхідні і достатні умови оптимальності із залученням других похідних функції Лагранжа за x , а саме, перевіряючи знакосталість матриці Гессе функції $L(x, \lambda_0, \Lambda)$.

З часів Лагранжа майже протягом цілого століття правило множників формували з $\lambda_0 = 1$, хоча без додаткових припущень (типу лінійної незалежності) у такому вигляді воно не є правильним.

Наприклад, розглянемо таку задачу:

$$x_1 \rightarrow \min, \quad x_1^2 + x_2^2 = 0.$$

Очевидним і єдиним розв’язком задачі є $x = (0, 0)$, бо це єдина допустима точка.

Складемо функцію Лагранжа з $\lambda_0 = 1$ і застосуємо алгоритм:

$$L = x_1 + \lambda(x_1^2 + x_2^2),$$

звідки

$$\begin{cases} L'_{x_1} = 2\lambda x_1 + 1 = 0, \\ L'_{x_2} = 2\lambda x_2 = 0, \end{cases} \Rightarrow \begin{cases} x_1 = -1/(2\lambda), \\ x_2 = 0, \end{cases}$$

і якщо $\lambda \neq 0$, то умова $x_1 = -1/(2\lambda)$ несумісна з обмеженням зв’язку $x_1^2 + x_2^2 = 0$. Умова лінійної незалежності тут не виконується, а саме:

$$\nabla g_1(x) = 2 \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = 2 \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

Позначимо через

$$L''_{xx}(x, \lambda_0, \Lambda) = \lambda_0 \nabla^2 f(x) + \sum_{k=1}^s \lambda_k \nabla^2 g_k(x)$$

матрицю других частинних похідних функції Лагранжа за координатами вектора x .

Теорема 4.2. Нехай функції $f(x), g_k(x), 1 \leq k \leq s$ двічі диференційовні в точці $x^* \in \mathbb{R}^N$ і неперервно диференційовні в деякому її околі, причому градієнти $\nabla f(x), \nabla g_k(x), 1 \leq k \leq s$ є лінійно незалежними. Якщо x^* – локальний розв'язок задачі (4.1), (4.2), то

$$(L''_{xx}(x^*, \lambda_0^*, \Lambda^*)h, h) \geq 0$$

за будь-яких значень λ_0^* і $\Lambda^* = (\lambda_1^*, \dots, \lambda_s^*)$, які задовольняють умову (4.4) і усіх $h \in \mathbb{R}^N$, при яких

$$(\nabla g_k(x), h) = 0, \quad 1 \leq k \leq s.$$

Теорема 4.3. Нехай функції $f(x), g_k(x), 1 \leq k \leq s$ двічі неперервно диференційовні в точці $x^* \in \mathbb{R}^N$, яка задовольняє умови $g_i(x) = 0, 1 \leq i \leq s$. Якщо за деяких λ_0^* і $\Lambda^* = (\lambda_1^*, \dots, \lambda_s^*)$ має місце співвідношення (4.4) і, крім того,

$$(L''_{xx}(x^*, \lambda_0^*, \Lambda^*)h, h) > 0$$

для усіх ненульових $h \in \mathbb{R}^N$, при яких

$$(\nabla g_k(x), h) = 0, \quad 1 \leq k \leq s.$$

Тоді x^* – строгий локальний розв'язок задачі (4.1), (4.2).

Теорема 2 є необхідною умовою другого порядку наявності екстремуму в точці x^* , а теорема 3 – достатня умова локального розв'язання задачі.

У найпростіших випадках сформульовані теореми 1–3 дозволяють розв'язати завдання у явному вигляді.

Відтак, опишемо алгоритм визначення екстремальних точок методом множників Лагранжа.

1. Складають функцію Лагранжа;
2. Знаходять частинні похідні від функції Лагранжа за змінними x та λ і прирівнюють їх до нуля;
3. Розв'язують систему рівнянь (4.4), знаходять точки, в яких цільова функція задачі може мати екстремум;
4. Серед точок, підозрілих на екстремум, знаходять такі, в яких досягається екстремум, і обчислюють значення функції (4.1) у цих точках.

Метод множників Лагранжа можна застосовувати і коли додаткові обмеження задано у формі нерівностей. Алгоритм розв'язання задачі, якщо потрібно знайти екстремуми функції $f(x)$ за умови $g(x) \leq 0$, описано нижче.

1. Знаходимо стаціонарні точки безумовного екстремуму функції $f(x)$, використовуючи рівняння $f'_{x_i} = 0, i = 1, \dots, N$;

2. Серед знайдених точок відбираємо ті, що задовольняють нерівність $g(x) \leq 0$;

3. Знаходимо стаціонарні точки умовного екстремуму функції $f(x)$ за умови $g(x) = 0$ з такої системи рівнянь

$$f'_{x_i} + \lambda g'_{x_i} = 0, \quad i = 1, \dots, N;$$

$$g(x) = 0;$$

4. Отримані в п. 2 і 3 точки досліджують у подальшому так само, як це робили у задачі безумовного екстремуму.

Приклад 4.2. За планом виробництва продукції підприємству необхідно виготовити 180 штук виробів. Це можна робити двома технологічними способами. При виробництві I способом затрати становлять $4x_1 + x_1^2$ грош. од., а при виготовленні II способом вони складають $8x_2 + x_2^2$ грош. од. Визначити, скільки виробів потрібно виготовити кожним із способів, аби загальні витрати підприємства на виробництво продукції були мінімальними.

Розв'язання. Математична постановка задачі буде мати такий вигляд:

$$f(x) = 4x_1 + x_1^2 + 8x_2 + x_2^2 \rightarrow \min \quad (4.5)$$

за умов

$$x_1 + x_2 = 180, \quad (4.6)$$

$$x_1, x_2 \geq 0. \quad (4.7)$$

Для розв'язання задачі використаємо метод множників Лагранжа. Знайдемо мінімальне значення функції (4.5) за умов (4.6), тобто без урахування умови невід'ємності змінних (4.7). Для цього складемо функцію Лагранжа, а саме:

$$L(x_1, x_2, \lambda) = 4x_1 + x_1^2 + 8x_2 + x_2^2 + \lambda(180 - x_1 - x_2).$$

Обчислимо її частинні похідні і прирівняємо їх до нуля:

$$\begin{cases} L'_{x_1} = 4 + 2x_1 - \lambda = 0, \\ L'_{x_2} = 8 + 2x_2 - \lambda = 0, \\ L'_{\lambda} = 180 - x_1 - x_2 = 0. \end{cases}$$

Виключаючи з перших двох рівнянь λ , отримаємо систему двох рівнянь із двома змінними:

$$x_1 - x_2 = 2; \quad x_1 + x_2 = 180.$$

Розв'язавши її, маємо $x_1^* = 91, x_2^* = 89$. Матриця других частинних похідних функції $f(x)$ додатно визначена, отже, у точці $x^* = (91, 89)$ функція f

має умовний мінімум. Відтак, якщо підприємство виготовляє 91 виріб першим технологічним способом і 89 виробів другим технологічним способом, то загальні затрати будуть мінімальними і становитимуть 17278 грош. од.

Приклад 4.3. Виробник бляшаних консервних банок бажає максимізувати об'єм деякої партії банок при заданій площі A_0 жерсті, яка використовується.

Розв'язання. Побудуємо математичну модель задачі. Введемо змінні: r, h – радіус основи і висота однієї банки. Задача полягає у пошуку таких значень цих змінних, за яких функція

$$V(r, h) = \pi r^2 h$$

досягає свого максимального значення і при цьому виконуються такі обмеження:

$$A(r, h) = 2\pi r^2 + 2\pi r h = A_0.$$

Перший спосіб. Виразимо величину h через r (або, якщо це зручніше, величину r через h) і запишемо об'єм у вигляді функції тільки однієї змінної r . Відтак, обмеження на площу входить у вираз для об'єму. Далі проводимо дослідження першої та другої похідних отриманої функції для визначення характеру та положення екстремуму, тобто

$$h = \frac{A_0 - 2\pi r^2}{2\pi r};$$

$$V(r) = \frac{(A_0 - 2\pi r^2)r}{2};$$

$$V'(r) = \frac{A_0}{2} - 3\pi r^2;$$

$V'(r) = 0$, коли $\hat{r} = \sqrt{A_0/6\pi}$. Оскільки $V''(r) = -6\pi r$ і в критичній точці $V''(\sqrt{A_0/6\pi}) < 0$, то робимо висновок, що функція $V(r)$ в точці \hat{r} досягає свого локального максимуму. Підставимо \hat{r} у вираз для висоти банки і отримаємо оптимальне значення цього параметру:

$$\hat{h} = \sqrt{2A_0/3\pi}.$$

Цікаво отримати оптимальне відношення висоти до радіуса. Видно, що для отримання максимального об'єму висоту консервної банки слід зробити такою, що дорівнює діаметру (за умови заданої площі жерсті, що використовується).

Другий спосіб. Застосуємо метод множників Лагранжа. Для цього спочатку сформуємо функцію Лагранжа, тобто

$$L(r, h, \lambda) = V(r, h) + \lambda(A(r, h) - A_0);$$

де λ – множник Лагранжа. Цей вираз можна записати через параметри консервної банки таким чином:

$$L(r, h, \lambda) = \pi r^2 h + \lambda(2\pi r^2 + 2\pi r h - A_0).$$

Запишемо градієнт функції Лагранжа і прирівняємо кожен його компоненту до нуля. В результаті отримаємо систему рівнянь:

$$\begin{cases} \frac{\partial L(r, h, \lambda)}{\partial r} = 0; \\ \frac{\partial L(r, h, \lambda)}{\partial h} = 0; \\ \frac{\partial L(r, h, \lambda)}{\partial \lambda} = 0; \end{cases} \Leftrightarrow \begin{cases} 2\pi r h + \lambda(4\pi r + 2\pi h) = 0; \\ \pi r^2 + \lambda 2\pi r = 0; \\ 2\pi r^2 + 2\pi r h - A_0 = 0; \end{cases} \Leftrightarrow \begin{cases} r = -2\lambda; \\ h = 2r; \\ \lambda = \pm\sqrt{A_0/24\pi}. \end{cases}$$

Розв'яжемо її і одержимо такий результат:

$$r = 2\sqrt{A_0/24\pi}; \quad h = 4\sqrt{A_0/24\pi}.$$

Відмітимо, що для значення λ вибираються тільки від'ємні величини квадратного кореня, оскільки в саме цьому випадку r та h набувають фізично реальних значень. Відношення висоти до радіуса буде таким самим, як і отримане першим способом розв'язування задачі.

Рекомендована література та інформаційні ресурси

1. Ржевський С.В. Дослідження операцій / С.В. Ржевський, В.М.Александрова. – К.: Академвидав, 2006. – 560 с.
2. Математичні методи дослідження операцій: підручник / Є.А. Лавров, Л.П. Пертун, В.В. Шендрик та ін. – Суми : Сумський державний університет, 2017. – 212 с.
3. Нефьодов Ю.М. Методи оптимізації в прикладах і задачах : навчальний посібник / Ю.М. Нефьодов, Т.Ю. Балицька. – Київ : Кондор, 2011. – 324 с.
4. Методи оптимізації : Методичні вказівки до виконання лабораторних робіт для студентів напряму підготовки 6.050101 «Комп'ютерні науки», спеціальностей 8.05010102 «Інформаційні технології проектування» та 8.05010103 «Системне проектування» денної та заочної форм навчання / Укл. В.В. Бобін, В.В. Ладогубець, О.Д. Фіногенов. – К. : НТУУ «КПІ», 2011 р. – 128 с.
5. Оптимізація систем керування.: навч. посіб. для студ. спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» / Л. Р. Ладієва; КПІ ім. Ігоря Сікорського, 2020. – 211с.

Завдання для розв'язування у класі

1. Визначити умовні глобальні екстремуми функцій в поданих нижче задачах за допомогою методу безпосереднього виключення змінних і метода множників Лагранжа. Подати графічну ілюстрацію.

а) $x_1^2 + x_2^2 \rightarrow \text{extr}, x \in D, D = \{x \in \mathbb{R}^2 \mid x_1 + x_2 = 4\};$

б) $3x_1^2 + 2x_2^2 - 3x_1 + 1 \rightarrow \text{extr}, \quad x_1^2 + x_2^2 = 4;$

в) $(x_1 - 3)^2 + (x_2 - 5)^2 \rightarrow \text{extr}, x \in D,$

$$D = \{x \in \mathbb{R}^2 \mid -2x_1 + x_2 = 5; x_1^2 + x_2^2 \leq 10\};$$

г) $2(x_1 - 1)^2 + 3(x_2 - 3)^2 \rightarrow \text{extr}, x \in D,$

$$D = \{x \in \mathbb{R}^2 \mid x_1 + x_2 = 6; x_1 + x_2 \leq 10; x_1, x_2 \geq 0\}.$$

2. Використовуючи метод Лагранжа, визначити стаціонарні точки під час дослідження умовного екстремуму наведених нижче функцій.

а) $x_1 + x_2 + x_3 \rightarrow \text{extr}$ за умови $\frac{1}{x_1} + \frac{1}{x_2} + \frac{1}{x_3} = 1;$

б) $x_1 x_2 x_3 \rightarrow \text{extr},$

за умов $x_1 + x_2 + x_3 = 6, x_1 x_2 + x_1 x_3 + x_2 x_3 = 12.$

3. Визначити мінімум функції $f(x) = x_1^2 - 2x_1 x_2 + x_3^2$ за умов, що

$$x_1 + 2x_2 + x_3 = 1, 2x_1 - x_2 + x_3 = 5,$$

$$x_1, x_2, x_3 \geq 0.$$

Дослідити, наскільки зміниться величина мінімуму зі зміною вільних членів рівнянь зв'язку: першого на $+0,1$ і другого на $-0,5$?

4. На множині, визначеній нерівностями

$$x + 2y \leq 12,$$

$$x + y \leq 9,$$

знайти точки, в яких досягаються глобальні екстремуми таких нелінійних цільових функцій:

а) $z = (x - 2)^2 + (y - 3)^2;$

б) $z = 2(x - 5)^2 + (y - 7)^2;$

в) $z = (x - 7)(y - 1).$

Контрольні питання

1. Які є способи зведення задач умовної оптимізації до задач знаходження безумовного екстремуму?

2. До яких задач можна застосовувати метод множників Лагранжа?

3. У чому полягає сутність методу множників Лагранжа?

4. Яким чином складають функцію Лагранжа?

5. З яких етапів складається алгоритм визначення екстремальних точок методом множників Лагранжа?

Завдання для самостійної та індивідуальної роботи

Знайдіть стаціонарні точки функції $f(x)$ у задачі умовної оптимізації, якщо обмеження задано у вигляді $g(x) = 0$. Визначте їх характер.

Варіанти завдань

№ варіанта	$f(x)$	$g(x)$
1.	$x_1 + x_2$	$x_1^2 + x_2^2 - 4$
2.	$x_1^2 - x_1x_2 + x_2^2 - 6x_1 - x_2 - 2$	$-x_1 + x_2 - 7$
3.	$-x_2 + 6x_1 - 1$	$x_1^2 + x_2^2 - 1$
4.	$2x_1^2 - x_1x_2 + x_2^2 - 8x_1 + 8x_2 - 7$	$-x_1 + 2x_2 - 7$
5.	$-12x_1 + 3x_2 + 2$	$x_1^2 + x_2^2 - 9$
6.	$5x_1^2 - x_1x_2 + x_2^2 - x_1 + 8x_2 - 5$	$-x_1 + 3x_2 - 8$
7.	$-6x_1 + 8x_2$	$x_1^2 + x_2^2 - 16$
8.	$5x_1^2 - 6x_1x_2 + x_2^2 - 6x_1 + 8x_2$	$-x_1 + x_2 - 9$
9.	$3x_1 - 8x_2 - 4$	$x_1^2 + x_2^2 - 4$
10.	$3x_1^2 - 5x_2^2 - 6x_1 + 8x_2$	$-5x_1 + x_2 - 3$
11.	$-42x_1 + x_2$	$x_1^2 + x_2^2 - 1$
12.	$-2x_1^2 - x_2^2 - 6x_1 + 8x_2 - 7$	$-x_1 + x_2 - 7$
13.	$-6x_1 + x_2$	$x_1^2 + x_2^2 - 4$
14.	$8x_1^2 - x_1x_2 + x_2^2 - 6x_1 + 8x_2 - 7$	$-2x_1 + x_2 - 7$
15.	$6x_1 + 5x_2 - 9$	$x_1^2 + x_2^2 - 25$
16.	$-4x_1^2 - 16x_1x_2 - x_2^2 - 2x_1 + 8x_2$	$-2x_1 + x_2 - 4$
17.	$-2x_1 + 7x_2 - 1$	$x_1^2 + x_2^2 - 16$
18.	$9x_1^2 - 18x_1x_2 + x_2^2 - 6x_1 + 8x_2 - 1$	$-x_1 + 3x_2 - 5$
19.	$x_1 - x_2 - 7$	$x_1^2 + x_2^2 - 9$
20.	$7x_1^2 - x_1x_2 + x_2^2 - 6x_1 - 4x_2 - 2$	$-x_1 + 6x_2 - 7$
21.	$-x_1 + 3x_2$	$x_1^2 + x_2^2 - 4$
22.	$4x_1^2 - x_1x_2 + x_2^2 - 5x_1 + 18x_2$	$-x_1 + x_2 - 6$

№ варіанта	$f(x)$	$g(x)$
23.	$-6x_1 + 33x_2 - 17$	$x_1^2 + x_2^2 - 1$
24.	$8x_1^2 - x_1x_2 + 5x_2^2 + x_1 + 23x_2$	$-4x_1 + x_2 - 4$
25.	$-6x_1 + 4x_2$	$x_1^2 + x_2^2 - 16$
26.	$8x_1^2 - x_1x_2 + x_2^2 + 4x_1$	$-x_1 + 3x_2 - 3$
27.	$-x_1 + 6x_2$	$x_1^2 + x_2^2 - 9$
28.	$x_1^2 - 3x_1x_2 + x_2^2 - x_1 + 8x_2$	$-x_1 + 2x_2 - 7$
29.	$x_1 + 3x_2$	$x_1^2 + x_2^2 - 1$
30.	$8x_1^2 - 2x_1x_2 + x_2^2 - 5x_1 + 8x_2$	$-x_1 + x_2 - 7$

Додаткові завдання

Для заданої функції знайти найбільше і найменше значення на множині, що описується системою обмежень.

1. $F(x) = x_1^2 + x_2^2 - 4x_1 + 8x_2,$
 $x_1^2 + x_2^2 - 16 \leq 0, -x_1 + 2x_2 = 7.$

5. $F(x) = \frac{1}{x_1^2 - 2x_1x_2 + x_2^2},$
 $x_1 + 2x_2 = 10.$

2. $F(x) = x_1^2 - x_2^2 + 4x_2,$
 $x_1^2 - 36 \leq 0, x_1x_2 \geq 2.$

6. $F(x) = -x_1^2 + 2x_1 + x_2,$
 $x_1^2 + x_2^2 - 16 = 0.$

3. $F(x) = x_2^2 - 5x_1 + 10x_2,$
 $-x_1 + x_2 = 6, x_2 \geq 0.$

7. $F(x) = -10x_1 + x_2,$
 $x_2 - \frac{1}{x_1} \geq 2, |x_1 + 2x_2| \leq 10$

4. $F(x) = \frac{5}{x_1} - x_2,$
 $-3x_1 + 2x_2 = 12.$

8. $F(x) = x_1^2 - 6x_1x_2 + x_2^2 + 12x_1 + 18x_2,$
 $x_1^2 + x_2^2 - 9 = 0$

Практичне заняття № 5

Методи одновимірної оптимізації

Мета: вивчення і аналіз пошукових алгоритмів оптимізації функції однієї змінної: половинного поділу (діхотомії), золотого перетину, Фібоначчі.

План заняття

1. Постановка задачі одновимірної умовної оптимізації. Основна ідея методів половинного поділу, золотого перетину, Фібоначчі.
2. Опис і алгоритми методів

Опорні поняття

1. Постановка задачі одновимірної умовної оптимізації. Основна ідея методів половинного поділу, золотого перетину, Фібоначчі

Обчислювальні (наближені) методи оптимізації в дослідженні операцій використовуються для розв'язування задачі оптимізації, що базується на точному або наближеному обчисленні значення функції мети. У результаті із заданою точністю знаходять наближення до розв'язку задачі – точки екстремуму функції. Якщо вона не єдина, то шукають наближення до множини точок екстремуму.

Наближені методи оптимізації функції однієї змінної можна поділити на три групи:

- методи виключення інтервалів (метод поділу інтервалу навпіл, метод Фібоначчі, метод «золотого перетину»);
- методи поліноміальної інтерполяції (метод квадратичної інтерполяції та методи інтерполяції вищих порядків);
- методи з використанням похідних (метод середньої точки, метод хорд, метод Ньютона).

Будемо для визначеності розглядати далі так звані унімодальні функції $f(\gamma)$, тобто такі, що мають на заданому інтервалі $[a, b]$ єдиний мінімум. Не порушуючи спільності, вважатимемо, що $f(\gamma)$ мінімізується на інтервалі $[a, b]$; максимум функції знаходиться аналогічним способом або ж зведенням до задачі на мінімум, змінюючи знак цільової функції на протилежний.

Отже, задача одновимірної умовної оптимізації записується у такий спосіб:

$$f(\gamma) \rightarrow \min, \gamma \in [a, b].$$

Основна ідея таких методів, як половинного поділу, золотого перетину і Фібоначчі, полягає у побудові послідовності вкладених один в один відрізків, які мають одну єдину спільну точку, яка виявляється точкою оптимуму.

Розглянемо кожен з цих методів окремо.

2. Опис і алгоритми методів

Дихотомічний метод (метод половинного поділу або дихотомії). Він передбачає на кожному етапі обчислення значення функції у двох точках заданого інтервалу. При цьому з метою максимального зменшення інтервалу невизначеності на кожному кроці зазначені точки вибираються якомога ближче до середини інтервалу.

Припустимо, на першому етапі експеримент проводиться в точках $\gamma_1 = x - \frac{\varepsilon}{2}$ і $\gamma_2 = x + \frac{\varepsilon}{2}$, де $x = \frac{a+b}{2}$ – середина інтервалу $[a, b]$, ε – достатньо мале додатне число (це число можна інтерпретувати як чутливість експериментатора у розрізненні двох близьких точок).

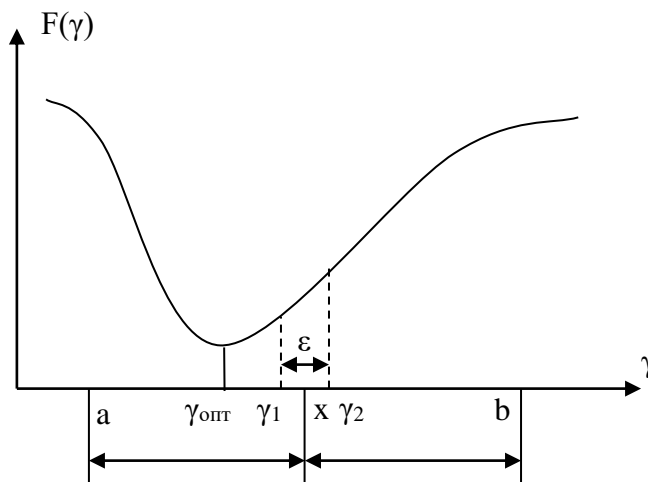


Рис. 5.1 Ілюстрація методу дихотомії

Якщо $f(x - \frac{\varepsilon}{2}) > f(x + \frac{\varepsilon}{2})$, тобто $f(\gamma_1) > f(\gamma_2)$, то з урахуванням унімодальності та можливості знаходження мінімуму в інтервалі ε для подальшого пошуку повинен бути залишений інтервал $[x - \frac{\varepsilon}{2}; b]$, в протилежному випадку – $[a; x + \frac{\varepsilon}{2}]$. Отже, якщо спочатку довжина інтервалу невизначеності L_0 дорівнює $(b - a)$, то після першого кроку вона стає такою (рис. 5.1):

$$L_1 = \frac{1}{2}(b - a) + \frac{\varepsilon}{2}.$$

Для отриманого інтервалу визначасмо нову пару точок: $\gamma_1 = x - \frac{\varepsilon}{2}$ і $\gamma_2 = x + \frac{\varepsilon}{2}$, де x – середина поточного інтервалу. Порівнюємо значення цільової функції в них і відповідним чином зменшуємо проміжок, довжина якого тепер буде такою:

$$L_2 = \frac{1}{4}(b - a) + \frac{3}{4}\varepsilon.$$

Легко зрозуміти, що після n таких порівнянь мінімум функції $\gamma_{\text{опт}}$ буде належати проміжку довжини

$$L_n = 2^{-n}(b - a) + (1 - 2^{-n})\varepsilon. \quad (5.1)$$

З формули (5.1) видно, що для зменшення інтервалу невизначеності, наприклад, в 100 разів, якщо знехтувати величиною ε , знадобиться 7 порівнянь.

Опишемо алгоритм методу половинного поділу

Ініціалізація. Задаємо початок a і кінець b інтервалу, величину $\varepsilon > 0$ (точність методу).

Крок 1. Обчислюємо середину інтервалу $[a, b]$, $x = \frac{a+b}{2}$.

Крок 2. Обчислюємо точки $\gamma_1 = x - \frac{\varepsilon}{2}$ і $\gamma_2 = x + \frac{\varepsilon}{2}$ і значення в них цілової функції $f_1 = f(\gamma_1)$ і $f_2 = f(\gamma_2)$.

Крок 3. Якщо $f_1 > f_2$, то $a = \gamma_1$, інакше $b = \gamma_2$.

Крок 4. Повторюємо кроки 1 – 3, допоки довжина інтервалу $[a, b]$ є більшою за 2ε .

Крок 5. Результат: $\gamma^{\text{опт}} \approx \frac{a+b}{2}$, $f^{\text{опт}} \approx f(\gamma^{\text{опт}})$.

Метод Фібоначчі є симетричною одновимірною оптимізацією, що вимагає мінімальної кількості обчислення значень функції. Тут на кожному кроці, крім першого, проводиться не два, а один розрахунок. Стратегія пошуку полягає в тому, що нова точка пошуку розташовується всередині інтервалу невизначеності симетрично щодо точки, яка вже знаходиться там і залишилася від попередніх порівнянь. Для визначення необхідного числа ітерацій n , яке забезпечує задану точність, і навіть для вибору положення перших перших точок пошуку, необхідно розглянути процес пошуку у зворотному порядку, тобто з останнього кроку.

Розглянемо ситуацію, яка склалася після того, як усі експерименти, окрім останнього, вже проведено. Довжину інтервалу невизначеності, що змінюється, позначимо L_{n-1} . Усередині цього інтервалу знаходиться точка з найменшим значенням функції $f(\gamma)$ (за результатами $(n - 1)$ порівнянь) і всередині нього також слід провести останнє порівняння. Очевидно, що для забезпечення мінімального інтервалу невизначеності L_n після n експериментів зазначені точки мають бути симетрично розташовані щодо середини інтервалу L_{n-1} і віддалені від неї на відстань $\frac{\varepsilon}{2}$ (рис. 5.2).

Отже,

$$L_{n-1} = 2L_n - \varepsilon. \quad (5.2)$$

Розглянемо далі ситуацію, коли проведено всі експерименти, крім двох останніх, і довжину наявного інтервалу невизначеності позначимо L_{n-2} . Усередині цього інтервалу знаходиться точка з найменшим із $(n - 2)$ випробувань значенням $F(\gamma)$, і також усередині нього необхідно провести наступний $(n - 1)$ експеримент. За його результатами цього експерименту

частина інтервалу L_{n-2} повинна бути відкинута, а частина, що залишилася, є L_{n-1} (рис. 5.3). Оскільки заздалегідь не зрозуміло, яка частина буде відкинута, ці точки мають розташовуватися на рівних відстанях від кінців інтервалу L_{n-2} .

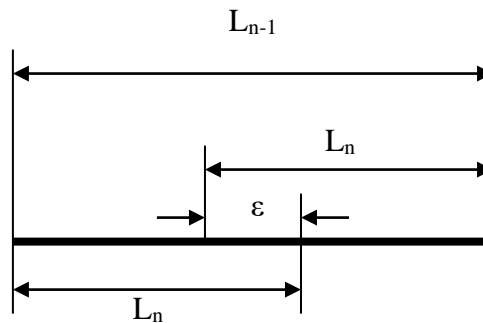


Рис. 5.2. Інтервали невизначеності на двох останніх ітераціях в методі Фібоначчі

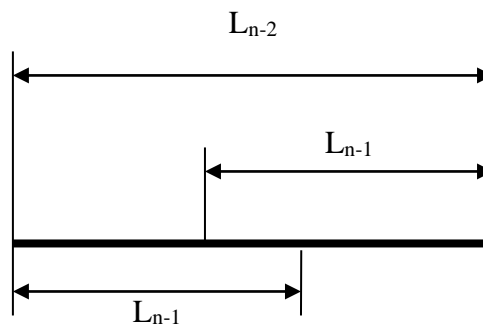


Рис. 5.3. Інтервали невизначеності на двох передостанніх ітераціях в методі Фібоначчі

Але одна з двох обраних точок всередині інтервалу L_{n-2} залишиться після $(n - 1)$ експерименту та стане однією з двох точок інтервалу L_{n-1} . Поєднання можливих комбінацій (рис. 5.2 та 5.3) приводить до схеми розбиття інтервалу L_{n-2} , зображених на рис. 5.4.

З рис. 5.4 випливає, що

$$L_{n-2} = L_{n-1} + L_n.$$

Аналогічно

$$L_{n-3} = L_{n-2} + L_{n-1}.$$

В загальному випадку

$$L_{j-1} = L_j + L_{j+1}, \quad j = \overline{2, n-1}. \quad (5.3)$$

Таким чином,

$$\begin{aligned} L_{n-1} &= 2L_n - \varepsilon, \\ L_{n-2} &= L_{n-1} + L_n = 2L_n - \varepsilon, \\ L_{n-3} &= L_{n-2} + L_{n-1} = 5L_n - 2\varepsilon, \\ L_{n-4} &= L_{n-3} + L_{n-2} = 8L_n - 3\varepsilon. \end{aligned}$$

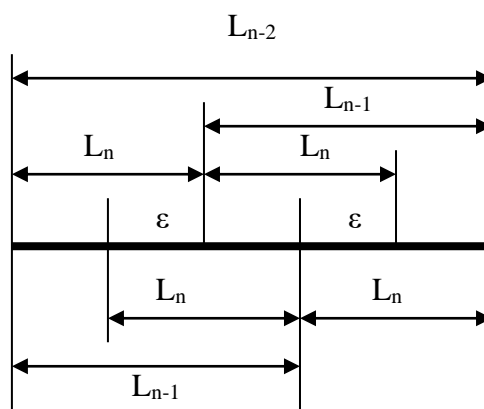


Рис. 5.4. Схема розбиття інтервалів в методі Фібоначчі

Для отримання загальної формули для довжини інтервалу L_{n-k} введемо послідовність чисел Фібоначчі F_k , яка задається наступною рекурентною формулою:

$$\begin{aligned} F_0 &= F_1 = 1, \\ F_k &= F_{k-1} + F_{k-2}, \quad k = 2, 3, \dots \end{aligned} \quad (5.4)$$

Тоді маємо

$$L_{n-k} = F_{k+1}L_n - F_{k-1}\varepsilon. \quad (5.5)$$

Враховуючи те, що після першого випробування довжина інтервалу невизначеності дорівнює $L_1 = (b - a)$, то у разі $k = n - 1$, отримуємо:

$$F_n L_n - F_{n-2} \varepsilon = L_1.$$

Звідси

$$L_n = \frac{L_1}{F_n} + \frac{F_{n-2}}{F_n} \varepsilon. \quad (5.6)$$

Відтак, після n випробувань початковий інтервал невизначеності, якщо знехтувати величиною ε , зменшується в F_n разів. Для його зменшення, наприклад, в 100 раз, має бути виконано 11 випробувань.

Для того, щоб почати пошук за методом Фібоначчі, необхідно визначити положення перших двох точок проведення випробувань. Ці точки розташовуються симетрично всередині початкового інтервалу невизначеності з відривом L_2 від відповідних кінців цього інтервалу. Вважаючи у виразі (5.5) $k = n - 2$, отримуємо

$$L_2 = F_{n-1}L_n - F_{n-3}\varepsilon = \frac{F_{n-1}L_1}{F_n} + \frac{F_{n-1}F_{n-2} - F_n F_{n-3}}{F_n} \varepsilon = \frac{F_{n-1}L_1}{F_n} + \frac{(-1)^n}{F_n} \varepsilon.$$

Нехтуючи величиною ε , маємо

$$L_2 = \frac{F_{n-1}}{F_n} L_1. \quad (5.7)$$

Опишемо алгоритм методу Фібоначчі.

Ініціалізація. Задані почіток a і кінець b інтервалу, точність методу ε .
Нехай $L = (b - a)$ – довжина вихідного інтервалу.

Крок 1. Розраховуємо кількість ітерацій n , користуючись умовою:

$$F_{n+1} \leq \frac{b - a}{\varepsilon} < F_{n+2}$$

і формуємо масив чисел Фібоначчі.

Крок 2. $k = 1$.

Знаходимо дві точки $\gamma_1 = a + \frac{F_n}{F_{n+2}}L$ і $\gamma_2 = a + b - \gamma_1 = a + \frac{F_{n+1}}{F_{n+2}}L$ і обчислимо значення в них цільової функції $f_1 = f(\gamma_1)$ і $f_2 = f(\gamma_2)$.

Крок 3. Допоки $k < n$,

початок циклу

$$k := k + 1;$$

якщо $f_1 > f_2$, то

$$a = \gamma_1, \gamma_1 = \gamma_2; f_1 = f_2,$$

$$\gamma_2 = a + \frac{F_{n-k+1}}{F_{n+2}}L = a + (b - \gamma_1), f_2 = f(\gamma_2);$$

інакше

$$b = \gamma_2, \gamma_2 = \gamma_1, f_2 = f_1,$$

$$\gamma_1 = a + \frac{F_{n-k}}{F_{n+2}}L = b - (\gamma_2 - a), f_1 = f(\gamma_1).$$

кінець циклу

Крок 4. $k = n$. Результат $\gamma^{\text{опт}} \approx \gamma_1 = \gamma_2$, $f^{\text{опт}} \approx f_1 = f_2$.

Приклад 5.1. Знайти мінімум функції $f(x) = (x - 2)^2$ із заданою точністю $\varepsilon = 0,01$, коли $x \in [1,4]$;

Розв'язання. Спочатку обчислимо кількість ітерацій. Отже, $\frac{b-a}{\varepsilon} = 300 > F_{n+2} = F_{13} = 233$. Відтак, $n = 11$. Результати розрахунків за алгоритмом методу Фібоначчі показано в табл. 5.1.

Таблиця 5.1

Розрахунки за методом Фібоначчі

k	a	b	g1	g2	f(g1)	f(g2)
1	1	4	2.145923	2.854077	0.021293	0.729448
2	1	2.854077	1.708155	2.145923	0.085174	0.021293
3	1.708155	2.854077	2.145923	2.416309	0.021293	0.173313
4	1.708155	2.416309	1.978541	2.145923	0.00046	0.021293
5	1.708155	2.145923	1.875536	1.978541	0.015491	0.00046
6	1.875536	2.145923	1.978541	2.042918	0.00046	0.001842
7	1.875536	2.042918	1.939914	1.978541	0.00361	0.00046
8	1.939914	2.042918	1.978541	2.004292	0.00046	1.84E-05
9	1.978541	2.042918	2.004292	2.017167	1.84E-05	0.000295
10	1.978541	2.017167	1.991416	2.004292	7.37E-05	1.84E-05
11	1.991416	2.017167	2.004292	2.004292	1.84E-05	1.84E-05

Як можна помітити, на 11-й ітерації дві точки γ_1 і γ_2 збігаються. Отже, критерієм закінчення ітераційного процесу методу Фібоначчі є $k = n$.

Метод «золотого перетину». Для початку розв'язання задачі одновимірної умовної оптимізації за методом Фібоначчі необхідно задати число експериментів n виходячи з допустимого значення інтервалу невизначеності в кінці пошуку. Цього недоліку не має метод «золотого перетину», котрий майже такий самий ефективний як і метод Фібоначчі. Тут так само, як і в останньому, точка, що вибирається всередині інтервалу невизначеності для проведення експерименту на черговому кроці, розташовується симетрично щодо точки, що вже знаходиться там, оскільки залишилася від попереднього експерименту. Тому тут для трьох сусідніх інтервалів невизначеності також є справедливим співвідношення (5.3). Однак у методі "золотого перетину" не використовується співвідношення (5.2), яке залежить від n . Натомість витримується сталим відношення довжин послідовних інтервалів, яке дорівнює τ , тобто

$$\frac{L_{j-1}}{L_j} = \frac{L_j}{L_{j+1}} = \tau. \quad (5.8)$$

Розділимо (5.3) на L_{j+1} і прийнявши до уваги (5.8), отримуємо:

$$\frac{L_{j-1}}{L_{j+1}} = \tau^2.$$

Маємо

$$\tau^2 = \tau + 1, \quad (5.9)$$

звідки

$$\tau = \frac{(1+\sqrt{5})}{2} \approx 1,618033989. \quad (5.10)$$

Тоді

$$\frac{L_{j-1}}{L_{j+1}} = \tau^2; \quad \frac{L_{j-2}}{L_{j+1}} = \tau^3 \quad \text{і т.д.}$$

Відтак,

$$\frac{L_1}{L_n} = \tau^{n-1}, \text{ тобто } L_n = \frac{L_1}{\tau^{n-1}}. \quad (5.11)$$

Таким чином, у методі «золотого перетину» початковий відрізок ділиться за «правилом золотого перерізу» (5.8), (5.10) і перші дві точки розташовуються симетрично на відстані $\bar{\tau} \approx 0,618$ від відповідних кінців інтервалу:

$$\gamma_1 = a + \frac{(3-\sqrt{5})}{2}(b-a) \quad \text{і} \quad \gamma_2 = a + \frac{(\sqrt{5}-1)}{2}(b-a).$$

Далі порівнюються значення функції в цих точках і на наступну ітерацію залишається один з відрізків $[a, \gamma_2]$ або $[\gamma_1, b]$ і т.д., допоки довжина поточного інтервалу не буде меншою за задану точність.

Можна показати, що остаточний інтервал невизначеності в даному методі при досить великих n лише на 17 % більше, ніж у методі Фібоначчі. Крім того,

коли значення n велике, обидва методи починаються практично з однієї точки, оскільки

$$\lim_{n \rightarrow \infty} \frac{F_{n-1}}{F_{n+1}} = \frac{(3 - \sqrt{5})}{2}, \quad \lim_{n \rightarrow \infty} \frac{F_n}{F_{n+1}} = \frac{(\sqrt{5} - 1)}{2}.$$

Опишемо алгоритм методу золотого перетину

Ініціалізація. Задано початок a і кінець b інтервалу, точність методу ε . $L = (b - a)$ – довжина вихідного інтервалу. $k = 0$.

1. Знаходимо дві точки:

$$\gamma_1 = a + \frac{(3-\sqrt{5})}{2}(b-a) \quad \text{і} \quad \gamma_2 = a + \frac{(\sqrt{5}-1)}{2}(b-a)$$

та значення в них цільової функції $f_1 = f(\gamma_1)$ і $f_2 = f(\gamma_2)$.

2. Допоки $(b - a) > \varepsilon$,

початок циклу

$$k := k + 1;$$

якщо $f_1 > f_2$, то

$$a = \gamma_1, \gamma_1 = \gamma_2; f_1 = f_2,$$

$$\gamma_2 = a + \frac{(\sqrt{5}-1)}{2}(b-a) = a + (b - \gamma_1), f_2 = f(\gamma_2);$$

інакше

$$b = \gamma_2, \gamma_2 = \gamma_1, f_2 = f_1,$$

$$\gamma_1 = a + \frac{(3-\sqrt{5})}{2}(b-a) = b - (\gamma_2 - a), f_1 = f(\gamma_1).$$

кінець циклу

3. Результат $\gamma^{\text{опт}} \approx \gamma_1 = \gamma_2$, $f^{\text{опт}} \approx f_1 = f_2$, k – кількість ітерацій.

Рекомендована література та інформаційні ресурси

1. Ржевський С.В. Дослідження операцій / С.В. Ржевський, В.М.Александрова. – К.: Академвидав, 2006. – 560 с.
2. Математичні методи дослідження операцій: підручник / Є.А. Лавров, Л.П. Пертун, В.В. Шендрик та ін. – Суми : Сумський державний університет, 2017. – 212 с.
3. Нефьодов Ю.М. Методи оптимізації в прикладах і задачах : навчальний посібник / Ю.М. Нефьодов, Т.Ю. Балицька. – Київ : Кондор, 2011. – 324 с.
4. Бех О.В. Математичне програмування : навч. посібн. / О.В. Бех, Т.А. Городня, А.Ф. Щербак. – Львів : Магнолія-2006, 2014. – 200 с.
5. Оптимізація систем керування: навч. посіб. для студ. спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» / Л. Р. Ладієва; КПІ ім. Ігоря Сікорського, 2020. – 211с.

Контрольні питання

1. Поясніть поняття «оптимізація».
2. Які типи задач оптимізації існують?
3. Сформулюйте теорему Веєрштрасса для функції однієї змінної?
4. У чому полягає задача умовної оптимізації?
5. У якому випадку використовують методи одномірної оптимізації – «золотого перетину», Фібоначчі?
6. Сформулюйте задачу одномірної умовної (безумовної) оптимізації?
7. Дайте порівняльну характеристику методів одновимірної оптимізації.
8. В чому полягає сутність методів дихотомічного поділу, Фібоначчі, «золотого перетину».
9. Які ще методи одновимірної оптимізації існують, сформулюйте їхні ідеї і умови використання.

Завдання для самостійної та індивідуальної роботи

Для завдання відповідно до варіанта знайти положення точки екстремуму і екстремальні значення цільової функції $f(x)$ на інтервалі $[a, b]$ методами половинного ділення, «золотого перетину», Фібоначчі. Довжина кінцевого інтервалу не повинна перевищувати 0,0001. Дати порівняльну характеристику методів. Якщо інтервал не задано, вибрати його самостійно.

Порядок виконання роботи

1. Вивчити необхідний теоретичний матеріал.
2. Відповідно до свого варіанта вибрати функцію з таблиці і визначити для неї початковий інтервал пошуку локального екстремуму.
3. Для алгоритму Фібоначчі визначити кількість експериментів, що дозволяє зменшити інтервал невизначеності в 1000 разів.
4. Здійснити по 2 ітерації пошуку екстремуму заданої функції кожним з розглянутих методів, ілюструючи розрахунки графічно.
5. Розробити на ЕОМ програму, яка реалізовує кожен з розглянутих методів (мову програмування вибрати самостійно).
6. Отримати розв'язок задачі трьома методами за допомогою розробленої програми. Провести порівняльну характеристику результатів.
7. Оформити звіт про виконання завдання такого змісту:
 - постановка задачі;
 - результати, отримані в п. 3 – 5 порядку виконання роботи;
 - результати розв'язання і їх аналіз.

Варіанти завдань

Номер варіанта	Вид цільової функції $f(x)$	Границі інтервалу		Напрямок оптимізації
		a	b	
1.	$\frac{x^3}{3} + 2(e^{-x} - x^2 + 2x)$			max
2.	$x(\ln x - 1) - \left(\frac{1}{\pi}\right) \sin \pi x$	0,5	1,5	min
3.	$0.5e^x - 2x + \frac{x^3}{3} - 4$			min
4.	$x^4 - 14x^3 + 60x^2 - 70x$			min
5.	$-e^{-x} + 4\cos\left(\frac{x}{2}\right) + 1$			max
6.	$\frac{x^3}{3} - x^2 + x - 2\sin x$			min
7.	$\frac{1}{8}(\sin 2x - 2x) - e^{-x}$	0,5	1,5	max
8.	$x^4 + 4x^2 - 32x + 5$			min
9.	$\frac{2}{3}\sqrt{(x+1)^3} - 4\sin\left(\frac{\pi x}{2}\right)$			min
10.	$3x^4 - 16x^2 + 6x^2 + 72x$	-1,4	-0,4	min
11.	$-\frac{x^4}{3} + 2x^2 + 1$	-2,8	-1,8	max
12.	$\ln x - \frac{3}{\pi} \sin(\pi x)$			max
13.	$-\frac{x^3}{3} + e^x - 4x^2 + 5x$			max
14.	$0.4x^3 + 5\cos 4x - \frac{3}{\pi} \sin(\pi x)$			min
15.	$x^4 + 2x^2 - 128x + 45$	2,5	3,5	min
16.	$x \ln 2x - x + \frac{1}{\pi} \cos(\pi x)$	0,5	1,5	min
17.	$x + \cos 2x - 1$			max
18.	$4x - \operatorname{tg} x + 1$	0,2	1,2	max

Номер варіанта	Вид цільової функції $f(x)$	Границі інтервалу		Напрямок оптимізації
		a	b	
19.	$\frac{4\sqrt{x^3} - 1}{3} + e^{-x}$			min
20.	$\frac{1}{3}(x^3 + 2) + 0.2\cos(5x)$	0,2	1,2	min
21.	$2\ln x - \sin(\pi x)$			max
22.	$\frac{x^3 + 1}{3} + e^x - 2x$			min
23.	$3x^4 - 16x^3 - 24x^2 - 60x$	4,8	5,8	min
24.	$\frac{x^4}{4} - x^3 + 10$	2,3	3,3	min
25.	$\frac{10}{4x^3 - 9x^2 + 6x}$	0,2	1,2	max
26.	$\sqrt[3]{2x^3 + 3x^2 - 36x}$	1,7	2,7	min
27.	$x - \ln x + 2$	0,3	1,3	min
28.	$\frac{x^4}{x^3 - 1}$	1,1	2,1	min
29.	$\frac{4}{x} + \frac{1}{x^4}$	-1,4	-0,4	min
30.	$e^{-x} + e^{2x} + 3$	-1	3	min

Додаткові завдання (відрізок $[a, b]$ – довільний)

№	Функція	Екстремум
1	$F(x) = x^3 - 2x^2 + x - 1$	Min
2	$F(x) = -3x^2 + 4x - 2$	Max
3	$F(x) = 2(x - 2)^2 - 1$	Min
4	$F(x) = -x^3 + 3x^2 - 4$	Max
5	$F(x) = \frac{x}{1 + x^2}$	Min
6	$F(x) = -(x + 2)^2(x - 1)^2$	Max
7	$F(x) = x + \frac{1}{x}$	Min
8	$F(x) = e^x - 2x^2$	Max
9	$F(x) = (x - 1)^3(x + 1)^2$	Min
10	$F(x) = -e^{-x+x^2}$	Max
11	$F(x) = 2x^{-2}e^{3x}$	Min
12	$F(x) = -x^4 + 14x^3 - 60x^2 + 70x$	Max

Практичне заняття № 6

Основна задача опуклого програмування. Критерій оптимальності. Розв'язання задачі умовної оптимізації квадратичної функції

Мета: вивчення поняття опуклої множини, опуклої функції, необхідних і достатніх умов оптимальності для задачі опуклого програмування.

План заняття

1. Поняття опуклої множини і опуклої функції. Властивості функцій, заданих на опуклих множинах
2. Постановка задачі опуклого програмування
3. Функція Лагранжа. Поняття сідлової точки функції Лагранжа
4. Теорема Куна – Такера та її різні узагальнення
5. Застосування теореми Куна – Такера для розв'язання задачі квадратичного програмування

Опорні поняття

1. Поняття опуклої множини і опуклої функції. Властивості функцій, заданих на опуклих множинах

Серед завдань нелінійного програмування виділяється достатньо широкий клас задач опуклого програмування. В них і допустима множина, і цільова функція є опуклими. А також особливо легко вирішуються питання існування та єдиності розв'язків.

Визначення 6.1. Множина Ω називається опуклою, якщо разом із будь-якими своїми точками вона містить і весь відрізок, що їх з'єднує.

Тобто для довільних $x, y \in \Omega$ і $\lambda \in [0, 1]$ справедливо $\lambda x + (1 - \lambda)y \in \Omega$. Порожню множину вважатимемо опуклою за визначенням.

Опуклими множинами є, наприклад:

– будь-який лінійний підпростір H лінійного простору L . Дійсно, для довільних елементів $x, y \in H$ лінійному підпростору H , за означенням, належить і будь-яка лінійна комбінація $\alpha_1 x + \alpha_2 y$ цих елементів. Зокрема, таке твердження є правильним і для $\alpha_1 = \lambda, \alpha_2 = 1 - \lambda$, де $\lambda \in [0, 1]$;

– відрізок $\Omega = [x_1, x_2]$ в лінійному просторі L ;

– одноточкова множина, як окремий випадок відрізка;

– множина розв'язків системи лінійних алгебраїчних рівнянь $Ax = b$, де A – матриця розмірності $m \times n$, x – матриця-стовпчик розмірності n , а b – матриц-стовпчик розмірності m ;

– множина $E = \{x \in L: \|x - a\| \leq r\}$, де L – нормований простір, $\|x\|$ – норма елемента x в L , $a \in L$;

– гіперплощина $E = \{x \in L: (x, a) = b\}$ для будь-яких $a \in L$ і $b \in R$;

– та інші.

Визначення 6.2. Функція $f: R^m \rightarrow R$ називається опуклою, якщо для всіх $x, y \in R^m$ і $\lambda \in [0, 1]$ виконується нерівність:

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y). \quad (6.1)$$

Якщо умова (6.1) виконується як строга нерівність для $x \neq y$ і $\lambda \in (0, 1)$, то функція f називається строго опуклою.

Геометрично опуклість означає, що графік функції на інтервалі (x, y) , що з'єднує будь-які точки x і y , лежить не вище за пряму, що з'єднує точки $(x, f(x))$ та $(y, f(y))$ (див. рис. 6.1, а).

Функція f сильно опукла (з константою $c > 0$), якщо нерівність (6.1) виконується у наступній сильнішій формі:

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y) - \frac{c}{2} \lambda(1 - \lambda) \|x - y\|^2. \quad (6.2)$$

Геометрично це поняття можна інтерпретувати у такий спосіб. Нехай точки відрізка $[x, y]$, який з'єднує точки x та y , параметризовані (з параметром λ): $\lambda \rightarrow \lambda x + (1 - \lambda)y$. Права частина нерівності (6.2) визначає на цьому відрізку поліном другого порядку від λ . Графік сильно опуклої функції над відрізком $[x, y]$ лежить нижче за параболу, яка є графіком цього полінома (див. рис. 6.1, б).

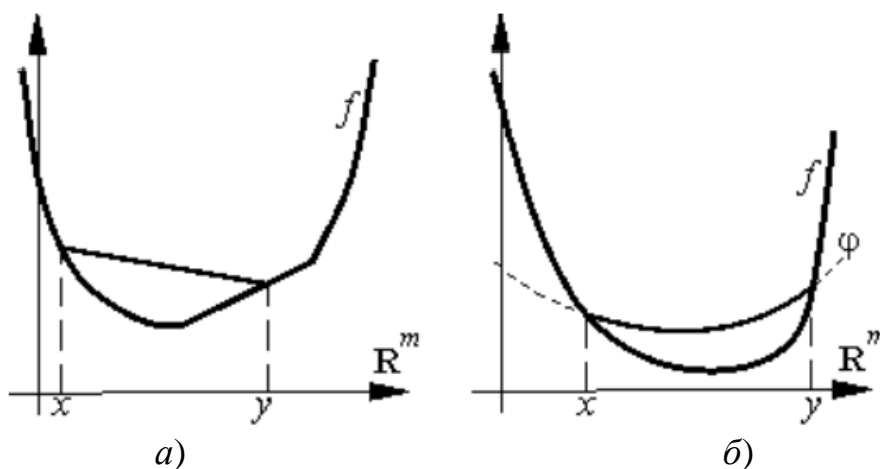


Рис. 6.1. Приклади опуклої (а) і сильно опуклої (б) функцій

Вочевидь для визначеної на множині X опуклої функції $f(x)$ множина $\{x \in X: f(x) \leq c\}$ також є опуклою (можливо порожньою). Звідси випливає, що у випадку, коли функції $f_1(x), f_2(x), \dots, f_k(x)$ визначені на опуклій множині X і опуклі, функція $f(x) = \min \{f_1(x), f_2(x), \dots, f_k(x)\}$ також опукла.

Опуклими є, наприклад:

– лінійна функція $f(x) = \sum_{i=1}^n a_i x_i$, де $x = (x_1, x_2, \dots, x_n)$, $a_i \in R$, $i = 1, 2, \dots, n$;

– функція $f(x) = \|x\|$, де $\|x\|$ – будь-яка норма в просторі R^n (октаедрична, Евклідова, Чебишева та інш.);

– будь-яка лінійна комбінація опуклих функцій $f_i(x), i = 1, 2, \dots, k$, визначених на опуклій множині $\Omega \subset R^n$ з невід’ємними коефіцієнтами $\alpha_i \geq 0, i = 1, 2, \dots, k$, а саме:

$$f(x) = \sum_{i=1}^k \alpha_i f_i(x), x \in \Omega;$$

– сума опуклої (строго опуклої) та лінійної функцій є функцією опуклою (строго опуклою).

Якщо $\mu(x)$ – опукла функція на опуклій множині $\Omega \subset R^n$, а $h(t)$ – опукла неспадна функція однієї дійсної змінної, визначена, по крайній мірі, на множині $\mu(\Omega)$, то складна функція $f(x) = h(\mu(x))$ є опуклою функцією на множині Ω . Якщо додатково функція $h(t)$ зростає, а функція $\mu(x)$ є строго опуклою, то функція $f(x)$ буде строго опуклою на множині Ω .

Нехай функція $f: \Omega \rightarrow R$ визначена на множині $\Omega \subset R^n$. Множина $G(f) = \{(x, y) \in R^{n+1}: f(x) \leq y\}$ називається надграфіком функції $f(x)$. Нерівність (6.1) рівносильна твердженню, що надграфік функції є опуклою множиною.

Крім того, справедливою є така теорема.

Теорема 6.1. Для того щоб функція $f: \Omega \rightarrow R$, визначена на множині $\Omega \subset R^n$, була опуклою, необхідно і достатньо, аби для будь-яких елементів $x^i \in \Omega, i = 1, 2, \dots, k$, та довільних чисел $\alpha_i \geq 0, i = 1, 2, \dots, k, \sum_{i=1}^k \alpha_i = 1$, виконувалась нерівність Єнсена³

$$f\left(\sum_{i=1}^k \alpha_i x^i\right) \leq \sum_{i=1}^k \alpha_i f(x^i).$$

Сформулюємо критерій опуклості диференційовної функції.

Твердження 6.1. Аби диференційовна функція f була опуклою, необхідно і достатньо виконання для всіх $x, y \in R^m$ такої нерівності

$$f(x) - f(y) \geq (f'(y), x - y). \quad (6.3)$$

Доведення

Дійсно, визначимо на відрізку $[0, 1]$ функцію φ наступним чином:

$$\varphi(\lambda) = f(\lambda x + (1 - \lambda)y).$$

Очевидно, функція φ буде опуклою одночасно з функцією f . Крім того, легко показати, що

$$\varphi'(\lambda) = (f'(\lambda x + (1 - \lambda)y), x - y).$$

Нерівність (6.3) в нових позначеннях перепишеться у такому вигляді:

$$\varphi(1) - \varphi(0) \geq \varphi'(0),$$

³ І.Л. Єнсен (1859 – 1925) – данський математик

або, якщо скористатися формулою Лагранжа,

$$\varphi'(\tau) \geq \varphi'(0), \quad (6.4)$$

де τ — деяка точка з інтервалу $(0, 1)$. З курсу математичного аналізу відомо, що для диференційовних функцій опуклість еквівалентна монотонності похідної. Тому, якщо f опукла, то $\varphi'(\lambda)$ монотонна. Отже, має місце еквівалентна формулі (6.3) нерівність (6.4).

Геометрично доведене твердження означає, що значення функції $f(x)$ "знаходяться вище" гіперплощини

$$H_y = \{(x, \xi) \in R^m \times R: \xi = f(y) + (f'(y), x - y)\},$$

яка є дотичною в точці $(y, f(y))$ до графіка

$$Gr f = \{(x, \xi) \in R^m \times R: \xi = f(x)\}$$

для всіх $y \in R^m$ (див. рис. 6.2).

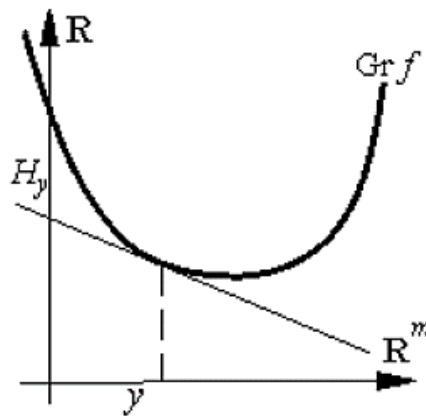


Рис. 6.2. Геометрична інтерпретація критерію опуклості диференційовної функції

Строга опуклість диференційовної функції еквівалентна строгій при $x \neq y$ нерівності (6.3). Сильна опуклість функції f еквівалентна виконанню при всіх x і y нерівності

$$f(x) - f(y) \geq (f'(y), x - y) + c\|x - y\|^2.$$

Зауваження. Функція $f \in C^2$ є сильно опуклою з константою c у тому і тільки у тому випадку, коли $f''(x) \geq c$ для всіх $x \in R^m$. Тобто, якщо функція двічі диференційовна, то для перевірки її на опуклість можна дослідити її матрицю Гессе на додатну (невід'ємну) визначеність.

Приклад 6.1. З'ясуємо, чи існує опукла множина $\Omega \subset R^2$, на якій функція

$$f(x_1, x_2) = x_1^3 + x_2^3 - x_1x_2 + 1$$

є строго опуклою.

Запишемо матрицю Гессе цієї функції

$$H = \begin{pmatrix} 6x_1 & -1 \\ -1 & 6x_2 \end{pmatrix}.$$

Для відповіді на поставлене питання достатньо вказати множину, на якій матриця Гессе є додатно визначеною. Згідно з критерієм Сильвестра для цього потрібно виконання двох умов:

$$\Delta_1 = 6x_1 > 0, \quad \Delta_2 = \det H = 36x_1x_2 - 1 > 0.$$

Відповідна множина Ω , обмежена гіперболою $x_1x_2 = 1/36$, наведена на рис. 6.3. Оскільки при $x_1x_2 < 1/36$ визначник матриці Гессе є від'ємним, в області Ω_2 між двома гілками гіперболи гессіан є знаконебезначеним. В області Ω_3 в третьом квадранті $\Delta_1 < 0$ і $\Delta_2 > 0$, тобто матриця Гессе визначена від'ємно. Отже, Ω є максимальною областю строгої опуклості заданої функції.

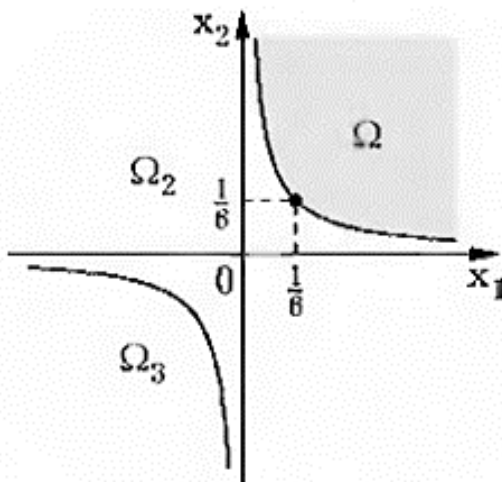


Рис. 6.3. Ілюстрація до прикладу 6.1

Розглянемо екстремальні властивості функцій, заданих на опуклих множинах.

Теорема 6.2. Нехай множина X – опукла, а функція $f(x)$ визначена і опукла на X . Тоді кожна точка локального мінімуму функції $f(x)$ одночасно є точкою її глобального мінімуму на X , причому множина

$$X_* = \{x: x \in X, f(x) = f_* = \min_{x \in X} f(x)\}$$

є опуклою. Якщо $f(x)$ строго опукла на X , то X_* містить не більше за одну точку мінімуму.

Теорема 6.3. Нехай множина X – опукла множина, а функція $f(x)$ визначена, неперервно диференційовна і опукла на X . Тоді, аби функція $f(x)$ була опуклою на X , необхідно і достатньо виконання нерівності

$$(f'(x), x - y) \geq f(x) - f(y) \geq ((f'(y), x - y) \forall x, y \in X.$$

Теорема 6.4. Нехай множина X – опукла, а функція $f(x)$ визначена, неперервно диференційовна на X , множина точок мінімуму $X_* = \{x: x \in X, f(x) = f_* = \min_{x \in X} f(x)\}$. Тоді в довільній точці $x_* \in X_*$ обов'язково виконується нерівність

$$(f'(x_*), x - x_*) \geq 0 \quad \forall x \in X. \quad (6.5)$$

У випадку $x_* \in \text{int}X_*$ нерівність (5) перетворюється на рівність $f'(x_*) = 0$. Якщо, крім того, $f(x)$ є опуклою на X , то умова (6.5) є достатньою, аби $x_* \in X_*$.

2. Постановка задачі опуклого програмування

У загальному випадку задачу опуклого програмування формулюють у такий спосіб:

$$f(x) \rightarrow \min_{x \in X}, \quad (6.6)$$

$$X = \{x \in D: g_i(x) \leq 0, \quad 1 \leq i \leq m; \quad g_i(x) = 0, \quad m+1 \leq i \leq s\},$$

$$D = \{x \in \mathbb{R}^n: x_i \geq 0, \quad 1 \leq j \leq n\},$$

де $f(x)$, $g_i(x)$, $1 \leq i \leq m$, – опуклі, $g_i(x)$, $m+1 \leq i \leq s$, – лінійні.

3. Функція Лагранжа. Поняття сідлової точки функції Лагранжа

Вище (практичне заняття 4) було встановлено, що множники Лагранжа можна використовувати при побудові критеріїв оптимальності для задач оптимізації з обмеженнями у вигляді рівностей. Кун і Такер узагальнили цей підхід на випадок загальної задачі нелінійного програмування з обмеженнями як у вигляді рівностей, так і нерівностей. Вони побудували необхідні і достатні умови оптимальності для задач нелінійного програмування, виходячи з припущення про диференційованість функцій.

Для задачі (6.6) побудуємо функцію Лагранжа:

$$L(x, y) = f(x) + \sum_{i=1}^s y_i g_i(x),$$

де $y \in Y$, $Y = \{y \in \mathbb{R}^s: y_i \geq 0, \quad 1 \leq i \leq m\}$.

Точка (x^*, y^*) називається сідловою для функції $L: D \times Y \rightarrow R$, якщо для всіх пар $(x, y) \in D \times Y$ виконується така нерівність:

$$L(x^*, y) \leq L(x^*, y^*) \leq L(x, y^*).$$

Графічну ілюстрацію поняття сідлової точки показано на рис. 6.4. Якщо ця умова справедлива лише для x та y , які є достатньо близькими до x^* та y^* відповідно, то пара (x^*, y^*) буде *локальною сідловою точкою* функції Лагранжа.

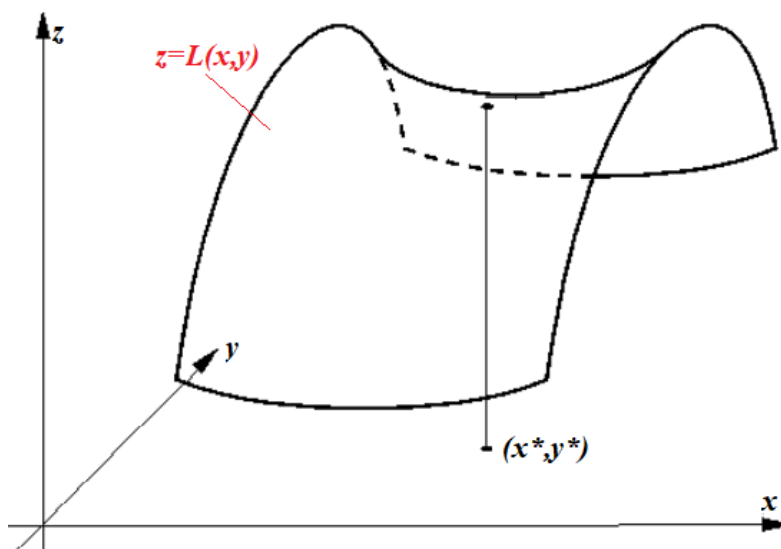


Рис. 6.4. Сідлова точка функції Лагранжа

4. Теорема Куна – Такера та її узагальнення

Визначення 6.1. Будемо говорити, що множина допустимих розв'язків задачі (6.6) задовольняє умову регулярності Слейтера, якщо існує принаймні одна точка $\hat{x} \in X$, для якої $g_i(\hat{x}) < 0$, для всіх $1 \leq i \leq m$.

Теорема 6.5. Нехай множина допустимих розв'язків задачі опуклого програмування (6.6) задовольняє умову регулярності. Тоді точка $x^* \in D$ є оптимальним розв'язком задачі (6.6) тоді і тільки тоді, коли існує такий вектор $y^* \in Y$, $Y = \{y \in \mathbb{R}^s: y_i \geq 0, 1 \leq i \leq m\}$, що пара (x^*, y^*) утворює сідлову точку функції Лагранжа.

Якщо вважати, що цільова функція $f(x)$ і функції $g_i(x)$ є неперервно диференційовними, то теорему Куна – Такера можна доповнити аналітичними виразами, котрі визначають необхідні і достатні умови того, що пара (x^*, y^*) складає сідлову точку функції L , тобто є розв'язком задачі (6.6). Ці вирази мають наступний вигляд:

$$\frac{\partial L}{\partial x_j} \geq 0 \quad \forall j = \overline{1, n}; \quad (6.7) \quad \frac{\partial L}{\partial y_i} \leq 0 \quad \forall i = \overline{1, m}; \quad (6.10)$$

$$\frac{\partial L}{\partial y_i} = 0 \quad \forall i = \overline{m+1, s}; \quad (6.11)$$

$$x_j \frac{\partial L}{\partial x_j} = 0 \quad \forall j = \overline{1, n}; \quad (6.8) \quad y_i \frac{\partial L}{\partial y_i} = 0 \quad \forall i = \overline{1, m}; \quad (6.12)$$

$$x_j \geq 0, \quad \forall j \in \overline{1, n}; \quad (6.9) \quad y_i \geq 0 \quad \forall i = \overline{1, m}. \quad (6.13)$$

Зауважимо, що всі частинні похідні функції Лагранжа в умовах (6.7) – (6.10) обчислюються в сідловій точці.

В табл. 6.1 наведено різні форми запису обмежень в задачі опуклого програмування і відповідні умови оптимальності Куна – Такера в диференціальній формі.

Таблиця 6.1.

Умови оптимальності Куна – Такера для задачі опуклого програмування залежно від форми запису обмежень

№	Обмеження, що описують допустиму множину X задачі (6.6)	Умови Куна – Такера	
1	$g_i(x) \leq 0, 1 \leq i \leq m;$ $g_i(x) = 0, m + 1 \leq i \leq s$ $x \in \mathbb{R}^n: x_i \geq 0, 1 \leq j \leq n$	$\frac{\partial L}{\partial x_j} \geq 0 \quad \forall j = \overline{1, n};$ $x_j \frac{\partial L}{\partial x_j} = 0 \quad \forall j = \overline{1, n};$ $x_j \geq 0 \quad \forall j = \overline{1, n};$	$\frac{\partial L}{\partial y_i} \leq 0 \quad \forall i = \overline{1, m};$ $\frac{\partial L}{\partial y_i} = 0 \quad \forall i = \overline{m + 1, s};$ $y_i \frac{\partial L}{\partial y_i} = 0 \quad \forall i = \overline{1, m};$ $y_i \geq 0 \quad \forall i = \overline{1, m}.$
2	$g_i(x) \leq 0, 1 \leq i \leq m;$ $g_i(x) = 0, m + 1 \leq i \leq s$ $x \in \mathbb{R}^n: x_i \geq 0, 1 \leq j \leq p,$ $p < n$	$\frac{\partial L}{\partial x_j} \geq 0 \quad \forall j = \overline{1, p};$ $\frac{\partial L}{\partial x_j} = 0 \quad \forall j = \overline{p + 1, n};$ $x_j \frac{\partial L}{\partial x_j} = 0 \quad \forall j = \overline{1, p};$ $x_j \geq 0 \quad \forall j = \overline{1, p};$	$\frac{\partial L}{\partial y_i} \leq 0 \quad \forall i = \overline{1, m};$ $\frac{\partial L}{\partial y_i} = 0 \quad \forall i = \overline{m + 1, s};$ $y_i \frac{\partial L}{\partial y_i} = 0 \quad \forall i = \overline{1, m};$ $y_i \geq 0 \quad \forall i = \overline{1, m}.$
3	$g_i(x) \leq 0, 1 \leq i \leq m;$ $x \in \mathbb{R}^n: x_i \geq 0, 1 \leq j \leq n$	$\frac{\partial L}{\partial x_j} \geq 0 \quad \forall j = \overline{1, n};$ $x_j \frac{\partial L}{\partial x_j} = 0 \quad \forall j = \overline{1, n};$ $x_j \geq 0 \quad \forall j = \overline{1, n};$	$\frac{\partial L}{\partial y_i} \leq 0 \quad \forall i = \overline{1, m};$ $y_i \frac{\partial L}{\partial y_i} = 0 \quad \forall i = \overline{1, m};$ $y_i \geq 0 \quad \forall i = \overline{1, m}.$
4	$g_i(x) = 0, 1 \leq i \leq s$ $x \in \mathbb{R}^n: x_i \geq 0, 1 \leq j \leq n$	$\frac{\partial L}{\partial x_j} \geq 0 \quad \forall j = \overline{1, n};$ $x_j \frac{\partial L}{\partial x_j} = 0 \quad \forall j = \overline{1, n};$ $x_j \geq 0 \quad \forall j = \overline{1, n};$	$\frac{\partial L}{\partial y_i} = 0 \quad \forall i = \overline{1, s};$

5	$g_i(x) \leq 0, 1 \leq i \leq m;$ $g_i(x) = 0, m + 1 \leq i \leq s$ $x \in \mathbb{R}^n$	$\frac{\partial L}{\partial x_j} = 0 \quad \forall j = \overline{1, n};$	$\frac{\partial L}{\partial y_i} \leq 0 \quad \forall i = \overline{1, m};$ $\frac{\partial L}{\partial y_i} = 0 \quad \forall i = \overline{m + 1, s};$ $y_i \frac{\partial L}{\partial y_i} = 0 \quad \forall i = \overline{1, m};$ $y_i \geq 0 \quad \forall i = \overline{1, m}.$
6	$g_i(x) = 0, 1 \leq i \leq s$ $x \in \mathbb{R}^n$	$\frac{\partial L}{\partial x_j} = 0 \quad \forall j = \overline{1, n};$	$\frac{\partial L}{\partial y_i} = 0 \quad \forall i = \overline{1, s}.$

5. Застосування теореми Куна – Такера для розв'язання задачі квадратичного програмування

Визначення 6.2. Квадратичною формою відносно змінних називається числова функція від цих змінних, яка записується у такий спосіб:

$$F = c_{11}x_1x_1 + c_{12}x_1x_2 + c_{13}x_1x_3 + \dots + c_{21}x_2x_1 + c_{22}x_2x_2 + \dots + c_{nn}x_nx_n =$$

$$= \sum_{k=1}^n \sum_{j=1}^n c_{kj}x_kx_j.$$

Визначення 6.3. Квадратична форма F називається додатно (від'ємно) визначеною, якщо $F(x) > 0$ ($F(x) < 0$) для всіх значень змінних $x = (x_1, x_2, \dots, x_n)$, крім $x = 0$.

Визначення 6.4. Квадратична форма F називається додатно (від'ємно) напіввизначеною, якщо $F(x) \geq 0$ ($F(x) \leq 0$) для будь-якого набору значень змінних $x = (x_1, x_2, \dots, x_n)$, і крім того, існує такий набір змінних $\hat{x} = (\hat{x}_1, \hat{x}_2, \dots, \hat{x}_n)$, де не всі значення змінних одночасно дорівнюють нулю, що $F(\hat{x})=0$.

Теорема 6.6. Квадратична форма являє собою опуклу функцію, якщо вона є додатно напіввизначеною, і увігнуту (опуклу догори) функцію, коли вона є від'ємно напіввизначеною.

Тепер розглянемо таку задачу квадратичного програмування: знайти максимальне (мінімальне) значення функції

$$f(x) = \sum_{j=1}^n d_j x_j + \sum_{k=1}^n \sum_{j=1}^n c_{kj} x_k x_j, \quad (6.13)$$

враховуючи обмеження

$$\sum_{j=1}^n a_{ij}x_j \leq b_i \quad (i = \overline{1, m}), \quad (6.14)$$

$$x_j \geq 0 \quad (j = \overline{1, n}), \quad (6.15)$$

де $\sum_{k=1}^n \sum_{j=1}^n c_{kj}x_kx_j$ – від’ємно (додатно) напіввизначена квадратична форма.

Для задачі (6.13) – (6.15) функцію Лагранжа записують у такому вигляді:

$$L(x, y) = \sum_{j=1}^n d_j x_j + \sum_{k=1}^n \sum_{j=1}^n c_{kj} x_k x_j + \sum_{i=1}^m y_i \left(b_i - \sum_{j=1}^n a_{ij} x_j \right).$$

Якщо функція L має сідлову точку (x^*, y^*) , то в цій точці виконуються наступні співвідношення:

$$\frac{\partial L}{\partial x_j} \leq 0 \quad \forall j = \overline{1, n}; \quad (6.16) \quad \frac{\partial L}{\partial y_i} \geq 0 \quad \forall i = \overline{1, m}; \quad (6.17)$$

$$x_j \frac{\partial L}{\partial x_j} = 0 \quad \forall j = \overline{1, n}; \quad (6.18) \quad y_i \frac{\partial L}{\partial y_i} = 0 \quad \forall i = \overline{1, m}; \quad (6.19)$$

$$x_j \geq 0 \quad \forall j = \overline{1, n}; \quad (6.20) \quad y_i \geq 0 \quad \forall i = \overline{1, m}. \quad (6.21)$$

Введемо додаткові змінні $v_j \geq 0 \quad j = \overline{1, n}$, і $w_i \geq 0 \quad i = \overline{1, m}$, які перетворюють нерівності (6.16) – (6.21) на рівності, і перепишемо вирази (6.16) – (6.21) таким чином:

$$\frac{\partial L}{\partial x_j} + v_j = 0 \quad (j = \overline{1, n}); \quad (6.22)$$

$$\frac{\partial L}{\partial y_i} - w_i = 0 \quad (i = \overline{1, m}); \quad (6.23)$$

$$x_j v_j = 0 \quad (j = \overline{1, n}); \quad (6.24)$$

$$y_i w_i = 0 \quad (i = \overline{1, m}); \quad (6.25)$$

$$x_j \geq 0, v_j \geq 0, y_i \geq 0, w_i \geq 0 \quad (j = \overline{1, n}, i = \overline{1, m}). \quad (6.26)$$

Відтак, аби знайти розв’язок задачі квадратичного програмування (6.13) – (6.15), потрібно визначити невід’ємний розв’язок системи лінійних рівнянь (6.22) і (6.23), яке задовольняє умови (6.24) і (6.25). Цей розв’язок можна знайти за допомогою методу штучного базису, який застосовується для пошуку максимального значення функції $F = \sum_i M y_i$ за умов (6.22), (6.23) (6.26) з урахуванням (6.24) та (6.25). Тут y_i – штучні змінні, які введені в рівняння (6.22), (6.23).

Використовуючи метод штучного базису і додатково враховуючи умови (6.24) та (6.25), після певної кількості кроків або встановимо нерозв'язність, або отримаємо оптимальний план вихідної задачі.

Отже, процес знаходження розв'язку задачі квадратичного програмування (6.13) – (6.15) включає наступні етапи:

1. Складаємо функції Лагранжа.
2. Записуємо у вигляді виразів (6.22) – (6.26) необхідні і достатні умови існування сідлової точки для функції Лагранжа.
3. Застосовуючи метод штучного базису або встановлюємо відсутність сідлової точки для функції Лагранжа, або знаходимо її координати.
4. Записуємо оптимальний розв'язок вихідної задачі і знаходимо значення цільової функції в цій точці.

Приклад 6.1. Знайти максимальне значення поданої нижче функції

$$f(x) = 2x_1 + 4x_2 - x_1^2 - x_2^2 \quad (6.27)$$

З огляду на такі обмеження

$$\begin{cases} x_1 + 2x_2 \leq 8, \\ 2x_1 - x_2 \leq 12, \end{cases} \quad (6.28)$$

$$x_1, x_2 \geq 0. \quad (6.29)$$

Розв'язання. Функція є увігнутою, оскільки складає суму лінійної функції $q = 2x_1 + 4x_2$ (яку можна вважати увігнутою) і квадратичної форми $z = -x_1^2 - x_2^2$, яка є від'ємно визначеною, а, отже, також увігнутою. Система обмежень задачі включає лише лінійні нерівності. Відтак, можна скористатися теоремою Куна – Такера. Складемо функцію Лагранжа, а саме:

$$L(x, y) = 2x_1 + 4x_2 - x_1^2 - x_2^2 + y_1(8 - x_1 - 2x_2) + y_2(12 - 2x_1 + x_2)$$

і запишемо у вигляді виразів (6.22) – (6.26) необхідні і достатні умови існування сідлової точки побудованої функції:

$$\begin{cases} \frac{\partial L}{\partial x_1} = 2 - 2x_1 - y_1 - 2y_2 \leq 0, \\ \frac{\partial L}{\partial x_2} = 4 - 4x_2 - 2y_1 + y_2 \leq 0, \\ \frac{\partial L}{\partial y_1} = 8 - x_1 - 2x_2 \geq 0, \\ \frac{\partial L}{\partial y_2} = 12 - 2x_1 + x_2 \geq 0; \end{cases} \quad (6.30)$$

$$\begin{cases} x_1 \frac{\partial L}{\partial x_1} = x_1(2 - 2x_1 - y_1 - 2y_2) = 0, \\ x_2 \frac{\partial L}{\partial x_2} = x_2(4 - 4x_2 - 2y_1 + y_2) = 0, \\ y_1 \frac{\partial L}{\partial y_1} = y_1(8 - x_1 - 2x_2) = 0, \\ y_2 \frac{\partial L}{\partial y_2} = y_2(12 - 2x_1 + x_2) = 0; \end{cases} \quad (6.31)$$

$$x_1, x_2, y_1, y_2 \geq 0. \quad (6.32)$$

Систему лінійних нерівностей (6.30) перепишемо у такий спосіб:

$$\begin{cases} 2x_1 + y_1 + 2y_2 \geq 2, \\ 4x_2 + 2y_1 - y_2 \geq 4, \\ x_1 + 2x_2 \leq 8, \\ 2x_1 - x_2 \leq 12. \end{cases} \quad (6.33)$$

Введемо тепер додаткові невід'ємні змінні v_1, v_2 і w_1, w_2 , які перетворюють нерівності (6.30) в рівності й отримаємо:

$$\begin{cases} 2x_1 + y_1 + 2y_2 - v_1 = 2, \\ 4x_2 + 2y_1 - y_2 - v_2 = 4, \\ x_1 + 2x_2 + w_1 = 8, \\ 2x_1 - x_2 + w_2 = 12. \end{cases} \quad (6.34)$$

$$x_1, x_2, v_1, v_2, w_1, w_2, y_1, y_2 \geq 0. \quad (6.35)$$

Враховуючи рівності (6.34), можна записати:

$$v_1 x_1 = 0, v_2 x_2 = 0, w_1 y_1 = 0, w_2 y_2 = 0. \quad (6.36)$$

Якщо знайти базисний розв'язок системи лінійних рівнянь (6.34) з урахуванням виконання умов (6.36), то буде отримана сідлова точка функції Лагранжа для вихідної задачі, тобто визначений оптимальний її розв'язок.

Для знаходження базисного розв'язку системи (34) скористаємося методом штучного базису. В перше і друге рівняння системи (34) відповідно додаємо додаткові невід'ємні змінні z_1 і z_2 і розглянемо задачу лінійного програмування:

$$F = -Mz_1 - Mz_2 \rightarrow \max, \quad (6.37)$$

за умов

$$\begin{cases} 2x_1 + y_1 + 2y_2 - v_1 + z_1 = 2, \\ 4x_2 + 2y_1 - y_2 - v_2 + z_2 = 4, \\ x_1 + 2x_2 + w_1 = 8, \\ 2x_1 - x_2 + w_2 = 12. \end{cases} \quad (6.38)$$

$$x_1, x_2, v_1, v_2, w_1, w_2, y_1, y_2, z_1, z_2 \geq 0. \quad (6.39)$$

В результаті розв'язання задачі (6.37) – (6.39) (при цьому враховуються умови (6.36) знайдемо допустимий базисний розв'язок системи лінійних рівнянь (6.38):

$$x_1 = 1, x_2 = 1; v_1 = v_2 = 0; w_1 = 5, w_2 = 11; y_1 = y_2 = 0.$$

Оскільки для цих значень виконуються умови (6.36), то $(\hat{x}, \hat{y}) = (1, 1, 0, 0)$ є сідловою точкою функції Лагранжа для вихідної задачі. Відтак, $\hat{x} = (1, 1)$ – оптимальний розв'язок, і $f_{max} = 3$.

Рекомендована література та інформаційні ресурси

1. Ржевський С.В. Дослідження операцій / С.В. Ржевський, В.М.Александрова. – К.: Академвидав, 2006. – 560 с.
2. Математичні методи дослідження операцій: підручник / Є.А. Лавров, Л.П. Пертун, В.В. Шендрик та ін. – Суми : Сумський державний університет, 2017. – 212 с.
3. Нефьодов Ю.М. Методи оптимізації в прикладах і задачах : навчальний посібник / Ю.М. Нефьодов, Т.Ю. Балицька. – Київ : Кондор, 2011. – 324 с.
4. Бех О.В. Математичне програмування : навч. посібн. / О.В. Бех, Т.А. Городня, А.Ф. Щербак. – Львів : Магнолія-2006, 2014. – 200 с.

Контрольні питання

1. Яка функція називається опуклою?
2. Що таке строго квазіопукла функція?
3. Сформулюйте критерій оптимальності першого порядку для задачі нелінійної умовної оптимізації.
4. Дайте визначення градієнта цільової функції.
5. Що таке гессіан функції?
6. Сформулюйте теорему Куна – Такера для задачі мінімізації опуклої функції з обмеженнями у вигляді рівностей.
7. Як записуються умови оптимальності теореми Куна – Такера для задачі мінімізації опуклої функції на невід'ємному ортанті?

Завдання для самостійної та індивідуальної роботи

Розв'язати задачу квадратичного програмування, застосовуючи теорему Куна – Такера

Номер варіанта	Завдання
1.	$f(x) = -x_1 + 2x_2 + 2x_1^2 + x_2^2 \rightarrow \min$ <p style="text-align: center;">за умов</p> $\begin{cases} x_1 + 2x_2 \leq 12, \\ 3x_1 + x_2 \leq 15, \\ x_1, x_2 \geq 0. \end{cases}$
2.	$f(x) = -6x_1 - 2x_2 + x_1^2 + x_2^2 \rightarrow \min$ <p style="text-align: center;">за умов</p> $\begin{cases} x_1 + x_2 \leq 18, \\ 2x_1 + x_2 \leq 10, \\ x_1, x_2 \geq 0. \end{cases}$
3.	$f(x) = 8x_1 + 4x_2 + x_1x_2 - 4x_1^2 - x_2^2 \rightarrow \max$ <p style="text-align: center;">за умов</p> $\begin{cases} x_1 + 2x_2 \leq 8, \\ 2x_1 - x_2 \leq 14, \\ x_1, x_2 \geq 0. \end{cases}$
4.	$f(x) = -x_1 + 3x_2 - 2x_1x_2 + x_1^2 + 8x_2^2 \rightarrow \min$ <p style="text-align: center;">за умов</p> $\begin{cases} x_1 + 3x_2 \leq 9, \\ -2x_1 + x_2 \leq 9, \\ x_1, x_2 \geq 0. \end{cases}$
5.	$f(x) = 12x_1 + 4x_2 - 6x_1^2 - x_2^2 \rightarrow \max$ <p style="text-align: center;">за умов</p> $\begin{cases} x_1 - 2x_2 \leq 8, \\ 3x_1 + x_2 \leq 12, \\ x_1, x_2 \geq 0. \end{cases}$
6.	$f(x) = -12x_1 - 24x_2 + 4x_1^2 + 9x_2^2 \rightarrow \min$ <p style="text-align: center;">за умов</p> $\begin{cases} -x_1 + 2x_2 \leq 8, \\ 2x_1 - x_2 \leq 10, \\ x_1, x_2 \geq 0. \end{cases}$

Номер варіанта	Завдання
7.	<p style="text-align: center;">$f(x) = 2x_1 + 4x_2 - x_1^2 - x_2^2 \rightarrow \max$</p> <p>за умов</p> $\begin{cases} x_1 + 2x_2 \leq 8, \\ 2x_1 - x_2 \leq 12, \\ x_1, x_2 \geq 0. \end{cases}$
8.	<p style="text-align: center;">$f(x) = -2x_1 + 4x_2 - 2x_1x_2 + x_1^2 + 8x_2^2 \rightarrow \min$</p> <p>за умов</p> $\begin{cases} x_1 + x_2 \leq 7, \\ 2x_1 - 3x_2 \leq 12, \\ x_1, x_2 \geq 0. \end{cases}$
9.	<p style="text-align: center;">$f(x) = -5x_1 + 8x_2 - x_1^2 - x_2^2 \rightarrow \max$</p> <p>за умов</p> $\begin{cases} x_1 + 3x_2 \leq 12, \\ 4x_1 + x_2 \leq 16, \\ x_1, x_2 \geq 0. \end{cases}$
10.	<p style="text-align: center;">$f(x) = -8x_1 - 5x_2 - x_1x_2 + x_1^2 + 4x_2^2 \rightarrow \min$</p> <p>за умов</p> $\begin{cases} 5x_1 - 2x_2 \leq 10, \\ x_1 + x_2 \leq 12, \\ x_1, x_2 \geq 0. \end{cases}$
11.	<p style="text-align: center;">$f(x) = 7x_1 + 10x_2 - x_1^2 - x_2^2 \rightarrow \max$</p> <p>за умов</p> $\begin{cases} x_1 + x_2 \leq 7, \\ 2x_1 - 3x_2 \leq 24, \\ x_1, x_2 \geq 0. \end{cases}$
12.	<p style="text-align: center;">$f(x) = -10x_1 - x_2 + x_1^2 + 8x_2^2 \rightarrow \min$</p> <p>за умов</p> $\begin{cases} x_1 + 2x_2 \leq 12, \\ 3x_1 + x_2 \leq 15, \\ x_1, x_2 \geq 0. \end{cases}$
13.	<p style="text-align: center;">$f(x) = x_1 + 8x_2 - x_1^2 - x_2^2 \rightarrow \max$</p> <p>за умов</p> $\begin{cases} x_1 + x_2 \leq 7, \\ x_2 \leq 5, \\ x_1, x_2 \geq 0. \end{cases}$

Номер варіанта	Завдання
14.	<p data-bbox="316 327 440 360">за умов</p> $f(x) = -5x_1 - 4x_2 + x_1^2 + 4x_2^2 \rightarrow \min$ $\begin{cases} x_1 + 2x_2 \leq 10, \\ -x_1 + x_2 \geq -8, \\ x_1, x_2 \geq 0. \end{cases}$
15.	<p data-bbox="316 808 440 842">за умов</p> $f(x) = 3x_1 + 6x_2 - x_1^2 - x_2^2 \rightarrow \max$ $\begin{cases} x_1 + 2x_2 \leq 8, \\ x_2 \leq 6, \\ x_1, x_2 \geq 0. \end{cases}$
16.	<p data-bbox="316 1066 440 1099">за умов</p> $f(x) = -2x_1 - 4x_2 - 2x_1x_2 + x_1^2 + 8x_2^2 \rightarrow \min$ $\begin{cases} x_1 + x_2 \leq 8, \\ 2x_1 - x_2 \leq 14, \\ x_1, x_2 \geq 0. \end{cases}$
17.	<p data-bbox="316 1099 440 1133">за умов</p> $f(x) = 7x_1 + 8x_2 - x_1^2 - 2x_2^2 \rightarrow \max$ $\begin{cases} 2x_1 + x_2 \leq 9, \\ 2x_1 - x_2 \leq 9, \\ x_1, x_2 \geq 0. \end{cases}$
18.	<p data-bbox="316 1346 440 1379">за умов</p> $f(x) = -9x_1 - 4x_2 + x_1^2 + x_2^2 \rightarrow \min$ $\begin{cases} x_1 + 2x_2 \leq 8, \\ 2x_1 + x_2 \leq 12, \\ x_1, x_2 \geq 0. \end{cases}$
19.	<p data-bbox="316 1603 440 1637">за умов</p> $f(x) = -2x_1 - 16x_2 + x_1^2 + 8x_2^2 \rightarrow \min$ $\begin{cases} x_1 + 2x_2 \leq 8, \\ x_1 - x_2 \leq -2, \\ x_1, x_2 \geq 0. \end{cases}$
20.	<p data-bbox="316 1861 440 1895">за умов</p> $f(x) = 8x_1 + 3x_2 - 4x_1^2 - x_2^2 \rightarrow \max$ $\begin{cases} 3x_1 + 2x_2 \leq 18, \\ 2x_1 - x_2 \leq -6, \\ x_1, x_2 \geq 0. \end{cases}$

Номер варіанта	Завдання
21. за умов	$f(x) = -12x_1 - x_2 + x_1^2 + x_2^2 \rightarrow \min$ $\begin{cases} x_1 + 2x_2 \leq 7, \\ 2x_1 - x_2 \leq -4, \\ x_1, x_2 \geq 0. \end{cases}$
22. за умов	$f(x) = 12x_1 + 4x_2 - 4x_1^2 - x_2^2 \rightarrow \max$ $\begin{cases} x_1 + 2x_2 \leq 6, \\ 2x_1 - x_2 \leq 10, \\ x_1, x_2 \geq 0. \end{cases}$
23. за умов	$f(x) = 2x_1 + 4x_2 - 2x_1x_2 + x_1^2 + 8x_2^2 \rightarrow \min$ $\begin{cases} 2x_1 + x_2 \leq 18, \\ x_1 + 3x_2 \leq 12, \\ x_1, x_2 \geq 0. \end{cases}$
24. за умов	$f(x) = -12x_1 - 14x_2 - x_1^2 - 7x_2^2 \rightarrow \max$ $\begin{cases} x_1 + 2x_2 \leq 8, \\ 3x_1 - x_2 \leq 7, \\ x_1, x_2 \geq 0. \end{cases}$
25. за умов	$f(x) = 36x_1 + 4x_2 - 4x_1x_2 + 9x_1^2 + 4x_2^2 \rightarrow \min$ $\begin{cases} x_1 + 2x_2 \leq 8, \\ -2x_1 + 3x_2 \leq -12, \\ x_1, x_2 \geq 0. \end{cases}$
26. за умов	$f(x) = 9x_1 + 5x_2 - x_1^2 - x_2^2 \rightarrow \max$ $\begin{cases} x_1 + 2x_2 \leq 10, \\ x_1 - x_2 \leq 6, \\ x_1, x_2 \geq 0. \end{cases}$
27. за умов	$f(x) = -2x_1 - 9x_2 + x_1^2 + x_2^2 \rightarrow \min$ $\begin{cases} 3x_1 + 2x_2 \leq 12, \\ 2x_1 + 5x_2 \leq 15, \\ x_1, x_2 \geq 0. \end{cases}$

Номер варіанта	Завдання	
28.	за умов	$f(x) = 22x_1 + 24x_2 - x_1^2 - x_2^2 \rightarrow \max$ $\begin{cases} 4x_1 + x_2 \leq 28, \\ 2x_1 - 3x_2 \leq 7, \\ x_1, x_2 \geq 0. \end{cases}$
29.	за умов	$f(x) = 7x_1 + 22x_2 - 5x_1x_2 + x_1^2 + 8x_2^2 \rightarrow \min$ $\begin{cases} 8x_1 + 2x_2 \leq 38, \\ 2x_1 - 7x_2 \leq 42, \\ x_1, x_2 \geq 0. \end{cases}$
30.	за умов	$f(x) = 9x_1 + 9x_2 - x_1^2 - x_2^2 \rightarrow \max$ $\begin{cases} x_1 + 2x_2 \leq 5, \\ 2x_1 - x_2 \leq 2, \\ x_1, x_2 \geq 0. \end{cases}$

Додаткові завдання

Розв'язати задачу квадратичного програмування, використовуючи, якщо це можливо, теорему Куна – Такера.

№	Функція	
1	за умов	$f(x) = x_1 + 14x_2 - 12x_1x_2 + x_1^2 + 8x_2^2 \rightarrow \min$ $\begin{cases} x_1 + 2x_2 \leq 18, \\ 2x_1 - x_2 \leq 12, \\ x_1, x_2 \geq 0. \end{cases}$
2	за умов	$f(x) = 12x_1 - 9x_2 - 4x_1^2 - x_2^2 \rightarrow \max$ $\begin{cases} -x_1 + 2x_2 \leq 14, \\ -2x_1 - x_2 \leq -5, \\ x_1, x_2 \geq 0. \end{cases}$
3	за умов	$f(x) = 2x_1 + 12x_2 + 3x_1x_2 + 5x_1^2 + x_2^2 \rightarrow \min$ $\begin{cases} x_1 - 2x_2^2 \leq 4, \\ 2x_1 + x_2 \leq 16, \\ x_1, x_2 \geq 0. \end{cases}$
4	за умов	$f(x) = -22x_1 + 18x_2 + 3x_1x_2 - x_1^2 - x_2^2 \rightarrow \max$ $\begin{cases} x_1 + 2x_2 \leq 28, \\ 2x_1 - x_2 = 55, \\ x_1 \geq 0. \end{cases}$

Практичне заняття № 7

Задачі дробово-лінійного програмування

Мета: засвоїти ідею і алгоритми методів розв'язання задач оптимізації дробово-лінійної функції за лінійних обмежень

План заняття

1. Постановка задачі дробово-лінійного програмування
2. Зведення задачі ДЛП до задачі лінійного програмування
3. Графічний метод розв'язання задачі ДЛП

Опорні поняття

1. Постановка задачі дробово-лінійного програмування

Нехай деяке підприємство виробляє n видів продукції. Позначимо p_j – прибуток від реалізації одиниці продукції j -го виду; c_j – собівартість одиниці продукції j -го виду; x_j – кількість продукції j -го виду, $j = 1, 2, \dots, n$; a_{ij} – витрати ресурсу i -го типу на виробництво одиниці продукції j -го виду; b_1, \dots, b_m – запаси ресурсів, які, за вимогою, потрібно використати повністю. Рентабельність витрат на виробництво може бути обчислена за такою формулою:

$$P(x) = \frac{p_1x_1 + p_2x_2 + \dots + p_nx_n}{c_1x_1 + c_2x_2 + \dots + c_nx_n}.$$

Зазначимо, що саме у вигляді частки визначається значна кількість економічних показників. Зазвичай, на змінні величини накладаються обмеження невід'ємності.

Отже, постановка задачі дробово-лінійного програмування (ДЛП) записується у такому вигляді:

$$F = \frac{p_1x_1 + p_2x_2 + \dots + p_nx_n}{c_1x_1 + c_2x_2 + \dots + c_nx_n} \rightarrow \text{extr}, \quad (7.1)$$

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1, \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2, \\ \dots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n = b_m, \end{cases} \quad (7.2)$$

$$x_j \geq 0, j = \overline{1, n}. \quad (7.3)$$

2. Зведення задачі ДЛП до задачі лінійного програмування

Проведемо заміну змінних за формулами:

$$y_0 = \frac{1}{c_1x_1 + c_2x_2 + \dots + c_nx_n}, \quad (7.4)$$

$$y_j = y_0x_j, j = 1, 2, \dots, n \quad (7.5)$$

Підставимо (7.4) – (7.5) в (7.1) :

$$F = y_0(p_1x_1 + \dots + p_nx_n),$$

$$F = p_1(y_0x_1) + \dots + p_n(y_0x_n).$$

Отже, цільова функція запишеться з новими змінними так:

$$F = p_1y_1 + p_2y_2 + \dots + p_ny_n.$$

Далі, підставляючи (7.5) в (7.2), маємо

$$\begin{cases} a_{11}\frac{y_1}{y_0} + a_{12}\frac{y_2}{y_0} + \dots + a_{1n}\frac{y_n}{y_0} = b_1, \\ \dots \\ a_{m1}\frac{y_1}{y_0} + a_{m2}\frac{y_2}{y_0} + \dots + a_{mn}\frac{y_n}{y_0} = b_m. \end{cases}$$

Оскільки $y_0 \neq 0$, тому є можливість помножити кожне рівняння на y_0 . Далі (7.4) можна подати у такому вигляді:

$$c_1 \cdot y_0x_1 + c_2 \cdot y_0x_2 + \dots + c_n \cdot y_0x_n = 1$$

або

$$c_1y_1 + c_2y_2 + \dots + c_ny_n = 1.$$

Остаточно від задачі дробово-лінійного програмування переходимо до задачі лінійного програмування відносно $(n+1)$ змінних, яку можна розв'язувати симплекс-методом:

$$\begin{cases} F = p_1y_1 + p_2y_2 + \dots + p_ny_n \rightarrow \max, \\ a_{11}y_1 + a_{12}y_2 + \dots + a_{1n}y_n - b_1y_0 = 0, \\ a_{21}y_1 + a_{22}y_2 + \dots + a_{2n}y_n - b_2y_0 = 0, \\ \dots \\ a_{m1}y_1 + a_{m2}y_2 + \dots + a_{mn}y_n - b_my_0 = 0, \\ c_1y_1 + c_2y_2 + \dots + c_ny_n = 1, \\ y_0 > 0, y_j \geq 0, j = 1, 2, \dots, n. \end{cases}$$

3. Графічний метод розв'язання задачі ДЛП

У випадку, коли $n = 2$, задачу можна розв'язати графічним методом. Отже, в цьому випадку маємо

$$f = \frac{p_1x_1 + p_2x_2 + p_0}{c_1x_1 + c_2x_2 + c_0} \rightarrow \text{extr}, \quad (7.6)$$

$$\begin{cases} a_{11}x_1 + a_{12}x_2 \leq b_1 \\ a_{21}x_1 + a_{22}x_2 \leq b_2 \\ \dots \\ a_{m1}x_1 + a_{m2}x_2 \leq b_m \end{cases} \quad (7.7)$$

$$x_j \geq 0, j = \overline{1,2}. \quad (7.8)$$

Нехай $p_0 = c_0 = 0$ і $c_1x_1 + c_2x_2 \neq 0$ (що не зменшує загальності). Розв'яжемо (7.6) відносно x_2 :

$$p_1x_1 + p_2x_2 = f(c_1x_1 + c_2x_2),$$

звідки

$$x_2 = \frac{p_1 - fc_1}{fc_2 - p_2}x_1; \quad x_2 = k \cdot x_1. \quad (7.9)$$

Рівняння (7.9) задає пряму з кутовим коефіцієнтом $k = k(f) = \frac{p_1 - fc_1}{fc_2 - p_2}$, яка проходить через початок координат. Знайдемо похідну:

$$\frac{dk}{df} = \frac{(p_1 - fc_1)' \cdot (fc_2 - p_2) - (fc_2 - p_2)'(p_1 - fc_1)}{(fc_2 - p_2)^2} = \frac{p_2c_1 - c_2p_1}{(fc_2 - p_2)^2}.$$

Знаменник дробу має сталий знак. Отже, знак похідної залежить від чисельника. Маємо два випадки.

1) Якщо $\Delta = p_2c_1 - c_2p_1 > 0$, то кутовий коефіцієнт k (функція) тільки зростає.

2) $\Delta = p_2c_1 - c_2p_1 < 0$, кутовий коефіцієнт k (функція) тільки спадає.

В обох випадках це відбувається при монотонному зростанні величини f (як аргумента функції). Звідси випливає, що в першому випадку f зростає при обертанні прямої проти годинникової стрілки, в другому випадку f зростає при обертанні прямої за годинниковою стрілкою.

Якщо $p_0 \neq 0$ і $c_0 \neq 0$, то центр пучка, через який проходять прямі при обертанні проти годинникової стрілки ($\Delta > 0$) або за годинниковою стрілкою ($\Delta < 0$), визначається з системи рівнянь:

$$\begin{cases} p_1x_1 + p_2x_2 + p_0 = 0, \\ c_1x_1 + c_2x_2 + c_0 = 0. \end{cases}$$

Приклад 7.1. Розв'язати подану нижче задачу дробово-лінійного програмування: а) графічним методом; б) використовуючи заміну змінних:

$$f = \frac{6x_1 + x_2 + 1}{11x_1 + 2x_2 + 1} \rightarrow \max, \quad \begin{cases} -2x_1 + 3x_2 \leq 9, \\ x_1 + x_2 \leq 6, \\ 2x_1 - x_2 \leq 10, \\ x_1, x_2 \geq 0. \end{cases}$$

Розв'язання. а) Будуємо множину допустимих розв'язків – багатокутник OABCD (рис.7.1) і визначаємо центр пучка з системи рівнянь

$$\begin{cases} 6x_1 + x_2 + 1 = 0 \\ 11x_1 + 2x_2 + 1 = 0 \end{cases} \Rightarrow x_1^o = -1; x_2^o = 5.$$

В результаті отримуємо, що центр пучка – точка $O_1(-1; 5)$. Визначимо напрямок зростання цільової функції. Для цього з'ясуємо знак виразу $p_2c_1 - c_2p_1$, а саме:

$$p_2c_1 - c_2p_1 = 1 \cdot 11 - 6 \cdot 2 = -1 < 0,$$

що означає зростання f за рухом годинникової стрілки. На рисунку бачимо, що максимум досягається в точці $O(0;0)$, причому $f_{\max} = f(0; 0) = -1$, а мінімум в точці $B(1.8;4.2)$, в якій $f_{\min} = f(1.8;4.2) = 0.547945$.

Зауважимо також, що в області допустимих розв'язків (всередині багатокутника) виконується нерівність $11x_1 + 2x_2 + 1 \neq 0$, оскільки ця лінія проходить поза п'ятикутником. Це означає, що в області допустимих розв'язків не відбудеться необмеженого зростання цільової функції.

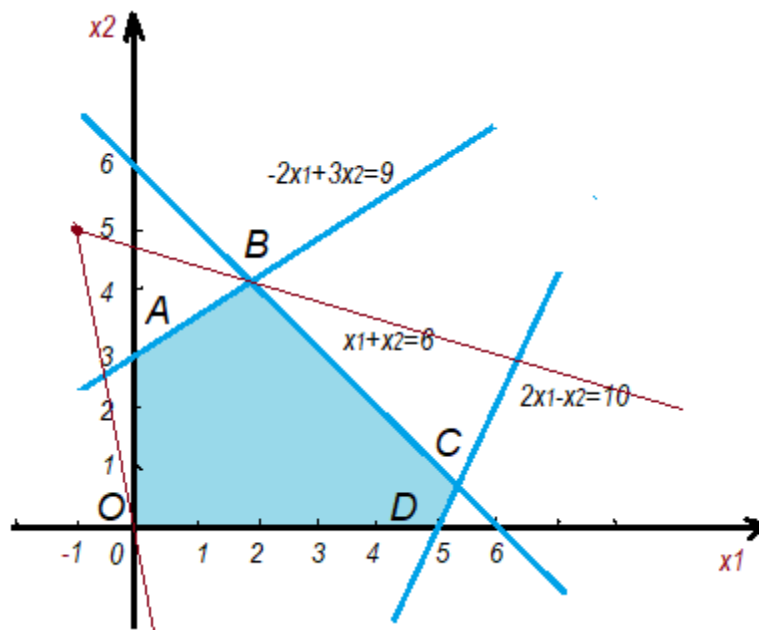


Рис. 7.1. Ілюстрація до графічного методу розв'язування задачі ДЛП

б) Перед тим як здійснити заміну змінних перейдемо до канонічної форми вихідної задачі, тобто

$$f = \frac{6x_1 + x_2 + 1}{11x_1 + 2x_2 + 1} \rightarrow \max, \quad \begin{cases} -2x_1 + 3x_2 + x_3 = 9 \\ x_1 + x_2 + x_4 = 6 \\ 2x_1 - x_2 + x_5 = 10 \end{cases}, \quad x_1, x_2, x_3, x_4, x_5 \geq 0.$$

Здійснюємо заміну

$$y_0 = \frac{1}{11x_1 + 2x_2 + 1} \quad y_j = y_0 x_j, \quad j = 1, 2, \dots, 5.$$

Тоді цільова функція набуває такого вигляду:

$$f = 6y_0 x_1 + y_0 x_2 + y_0 \rightarrow \max$$

або

$$f = 6y_1 + y_2 + y_0 \rightarrow \max$$

і відповідна система обмежень буде такою:

$$-2 \frac{y_1}{y_0} + 3 \frac{y_2}{y_0} + \frac{y_3}{y_0} = 9,$$

$$\frac{y_1}{y_0} + \frac{y_2}{y_0} + \frac{y_4}{y_0} = 6,$$

$$2 \frac{y_1}{y_0} - \frac{y_2}{y_0} + \frac{y_5}{y_0} = 10.$$

Помноживши ці рівняння на $y_0 \neq 0$ та додавши рівняння заміни, остаточно отримуємо задачу лінійного програмування:

$$f = 6y_1 + y_2 + y_0 \rightarrow \max,$$

$$\begin{cases} -2y_1 + 3y_2 + y_3 - 9y_0 = 0, \\ y_1 + y_2 + y_4 - 6y_0 = 0, \\ 2y_1 - y_2 + y_5 - 10y_0 = 0, \\ 2y_1 + 3y_2 + y_0 = 1, \end{cases}$$

$$y_j \geq 0, \quad j = \overline{0, 5},$$

яку можна розв'язати симплексним методом.

Скориставшись online калькулятором, знайдемо $y_1^* = 0$; $y_2^* = 0$; $y_3^* = 9$; $y_4^* = 6$; $y_5^* = 10$; $y_0^* = 1$ й $f_{\max} = 1$. Результати, отримані обома способами, збігаються.

Рекомендована література та інформаційні ресурси

1. Ржевський С.В. Дослідження операцій / С.В. Ржевський, В.М.Александрова. – К.: Академвидав, 2006. – 560 с.
2. Математичні методи дослідження операцій: підручник / Є.А. Лавров, Л.П. Пертун, В.В. Шендрік та ін. – Суми : Сумський державний університет, 2017. – 212 с.
3. Нефьодов Ю.М. Методи оптимізації в прикладах і задачах : навчальний посібник / Ю.М. Нефьодов, Т.Ю. Балицька. – Київ : Кондор, 2011. – 324 с.
4. Бех О.В. Математичне програмування : навч. посібн. / О.В. Бех, Т.А. Городня, А.Ф. Щербак. – Львів : Магнолія-2006, 2014. – 200 с.

Завдання для розв'язування

$$1. \quad F = \frac{6x_1 + 2x_2 + 23}{3x_1 - 5x_2 + 2} \rightarrow \max;$$

$$\begin{cases} x_1 + 2x_2 \leq 14, \\ -5x_1 + 3x_2 \leq 15, \\ 4x_1 + 6x_2 \geq 24, \\ x_1, x_2 \geq 0. \end{cases}$$

$$2. \quad F = \frac{6x_1 + 2x_2 + 5}{32x_1 - 7x_2 + 42} \rightarrow \max;$$

$$\begin{cases} 10x_1 + 3x_2 \geq 30, \\ -x_1 + x_2 \leq 5, \\ x_1 + x_2 \leq 10, \\ x_2 \geq 2, \\ x_1, x_2 \geq 0. \end{cases}$$

$$3. \quad F = \frac{-6x_1 + 2x_2 + 3}{3x_1 + 7x_2 + 2} \rightarrow \max;$$

$$\begin{cases} 4x_1 - 2x_2 \leq 12, \\ -x_1 + 3x_2 \leq 6, \\ 2x_1 + 4x_2 \geq 16, \\ x_1, x_2 \geq 0. \end{cases}$$

$$4. \quad F = \frac{16x_1 + 2x_2 - 52}{32x_1 + 7x_2 + 12} \rightarrow \min;$$

$$\begin{cases} 6x_1 + 6x_2 \geq 36, \\ 4x_1 + 8x_2 \geq 32, \\ 2x_1 \geq 2, \quad x_1, x_2 \geq 0. \end{cases}$$

$$5. \quad F = \frac{x_1 - 2x_2 + 3}{-3x_1 + 7x_2 + 12} \rightarrow \min;$$

$$\begin{cases} 3x_1 - 2x_2 \leq 12, \\ -x_1 + 2x_2 \leq 8, \\ 2x_1 + 3x_2 \geq 6, \\ x_1, x_2 \geq 0. \end{cases}$$

$$6. \quad F = \frac{7x_1 + 12x_2 + 35}{3x_1 + x_2 + 12} \rightarrow \max;$$

$$\begin{cases} x_1 + 2x_2 \leq 10, \\ x_1 + 2x_2 \geq 2, \\ 2x_1 + x_2 \leq 10, \\ x_1, x_2 \geq 0. \end{cases}$$

$$7. \quad F = \frac{6x_1 + 2x_2}{3x_1 + x_2 + 2} \rightarrow \min;$$

$$\begin{cases} 2x_1 + 4x_2 \leq 16, \\ -4x_1 + 2x_2 \leq 8, \\ x_1 + 3x_2 \geq 9, \\ x_1, x_2 \geq 0. \end{cases}$$

$$8. \quad F = \frac{-x_1 + 5x_2 + 15}{13x_1 + x_2 + 2} \rightarrow \min;$$

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

$$9. \quad F = \frac{6x_1 + 5x_2 + 30}{13x_1 + 7x_2 + 24} \rightarrow \max;$$

$$\begin{cases} 6x_1 + 6x_2 \leq 36, \\ 4x_1 + 2x_2 \leq 20, \\ 4x_1 + 8x_2 \leq 40, \\ x_1, x_2 \geq 0. \end{cases}$$

$$10. \quad F = \frac{-6x_1 + 2x_2 + 13}{3x_1 - 7x_2 + 21} \rightarrow \min;$$

$$\begin{cases} 5x_1 + 7x_2 \geq 35, \\ 2x_1 \geq 4, \\ x_2 \geq 1, \\ x_1, x_2 \geq 0. \end{cases}$$

Контрольні питання

1. Які властивості має задача дробово-лінійного програмування?
2. Дайте визначення опуклої множини?
3. Дайте економічну інтерпретацію цільової функції задачі ДЛП?
4. Як визначити напрямок зростання (спадання) цільової функції ЗДЛП?
5. Опишіть випадки, коли ЗДЛП не має розв'язків?
6. У яких випадках ЗДЛП має безліч розв'язків?
7. Як визначити лінійну комбінацію двох точок опуклої множини?
8. Яка множина на площині відповідає обмеженню-нерівності?
9. Чи може задача лінійного програмування мати три розв'язки?

Завдання для самостійної та індивідуальної роботи

Розв'язати задачу ДЛП за умови невід'ємності змінних: а) графічним методом, б) симплекс-методом, використавши метод зведення до задачі лінійного програмування.

Варіанти завдань

$$1. \begin{cases} x_1 + 3x_2 \leq 3, \\ 7x_1 + 5x_2 \leq 35, \\ 2x_1 + 4x_2 \leq -8, \\ -4x_1 + 6x_2 \leq 24, \end{cases}$$

$$\max Z = \frac{6x_1 + 2x_2}{3x_1 + 2x_2 + 4}$$

$$2. \begin{cases} x_1 + 5x_2 \geq 5, \\ 5x_1 + 15x_2 \leq 75, \\ -4x_1 + 5x_2 \geq -20, \\ x_1 - 2x_2 \geq -2, \end{cases}$$

$$\min Z = \frac{-3x_1 + 2x_2 + 2}{x_1 + 2x_2 + 4}$$

$$3. \begin{cases} x_1 + 12x_2 \geq 5, \\ -3x_1 + 3x_2 \geq -24, \\ 4x_1 + 10x_2 \leq 50, \\ 5x_1 - 12x_2 \geq -35, \end{cases}$$

$$\max Z = \frac{5x_1 + 3x_2 + 1}{3x_1 + x_2 + 3}$$

$$4. \begin{cases} 5x_1 - 2x_2 \geq -24, \\ x_1 + 2x_2 \geq 16, \\ x_1 - 4x_2 \leq -8, \\ 4x_1 + 3x_2 \leq 82, \end{cases}$$

$$\max Z = \frac{-7x_1 + 2x_2 - 5}{6x_1 + 8}$$

$$5. \begin{cases} x_1 + 6x_2 \geq 6, \\ x_2 \leq 3, \\ -\frac{3}{2}x_1 + \frac{9}{2}x_2 \geq -\frac{27}{4}, \\ 11x_1 + 7x_2 \leq 77, \end{cases}$$

$$\max Z = \frac{-6x_1 + 2x_2 + 5}{3x_1 + x_2 + 14}$$

$$6. \begin{cases} x_1 \leq 8, \\ 5x_1 + 7x_2 \geq 35, \\ -3x_1 + \frac{9}{2}x_2 \geq -\frac{27}{2}, \\ x_1 - 3x_2 \geq -12, \end{cases}$$

$$\min Z = \frac{-6x_1 + 2x_2 + 3}{3x_1 + 7x_2 + 2}$$

$$7. \quad \begin{cases} \frac{1}{2}x_1 + 2x_2 \geq 1, \\ 6x_1 + 10x_2 \leq 60, \\ -2x_1 + x_2 \geq -2, \\ -3x_1 + 5x_2 \leq 15, \end{cases}$$

$$\max Z = \frac{2x_2 + 3}{3x_1 + x_2 + 2}$$

$$8. \quad \begin{cases} 2x_1 + 4x_2 \geq 8, \\ 4x_1 + 8x_2 \leq 32, \\ x_1 \leq 7, \\ x_2 \leq 2, \end{cases}$$

$$\min Z = \frac{-6x_1 + 3}{x_1 + x_2 + 12}$$

$$9. \quad \begin{cases} -8x_1 + 5x_2 \leq 29, \\ 5x_1 + 2x_2 \leq 54, \\ x_1 + 4x_2 \geq 4, \\ -9x_1 + 3x_2 \geq -30, \end{cases}$$

$$\max Z = \frac{x_1 + 2x_2 + 7}{3x_1 + x_2 + 2}$$

$$10. \quad \begin{cases} 2x_1 + x_2 \geq 6, \\ 9x_1 + 6x_2 \leq 54, \\ \frac{7}{2}x_1 + \frac{9}{2}x_2 \geq -\frac{63}{4}, \\ 3x_1 - 4x_2 \geq -12, \end{cases}$$

$$\min Z = \frac{2x_2 + 32}{x_1 + x_2 + 2}$$

$$11. \quad \begin{cases} 2x_1 + 2x_2 \geq 4, \\ 8x_1 + 12x_2 \leq 96, \\ 3x_1 - 4x_2 \leq 12, \\ -5x_1 + 2x_2 \leq 10, \end{cases}$$

$$\max Z = \frac{-x_1 + 2x_2 + 8}{3x_1 - x_2 + 12}$$

$$12. \quad \begin{cases} -2x_1 + 2x_2 \geq -14, \\ x_1 + 3x_2 \geq 7, \\ x_1 - x_2 \leq 5, \\ x_2 \leq 3, \end{cases}$$

$$\max Z = \frac{-6x_1 + 12x_2 + 33}{3x_1 + 17x_2 + 12}$$

$$13. \quad \begin{cases} x_1 + 6x_2 \geq 6, \\ -\frac{1}{3}x_1 + 3x_2 \geq -1, \\ -x_1 + \frac{5}{2}x_2 \leq \frac{5}{2}, \\ 7x_1 + 11x_2 \leq 77, \end{cases}$$

$$\max Z = \frac{-6x_1 + x_2 - 3}{3x_1 + x_2 + 8}$$

$$14. \quad \begin{cases} x_1 \leq 6, \\ x_1 + 6x_2 \geq 6, \\ x_1 + 2x_2 \leq 12, \\ x_1 - x_2 \geq -1, \end{cases}$$

$$\max Z = \frac{x_1 - 2x_2 + 3}{x_1 + 7x_2 + 12}$$

$$15. \quad \begin{cases} -2x_1 + 2x_2 \geq -8, \\ -3x_1 + x_2 \geq -14, \\ 2x_1 + 2x_2 \geq 4, \\ -\frac{1}{2}x_1 + 3x_2 \leq \frac{21}{2}, \end{cases}$$

$$\min Z = \frac{x_1 + 2x_2 + 2}{x_1 + 7x_2 + 7}$$

$$16. \quad \begin{cases} x_1 \leq 6, \\ 2x_1 + 2x_2 \geq 4, \\ 8x_1 + 12x_2 \leq 96, \\ 3x_1 - 4x_2 \leq 12, \end{cases}$$

$$\min Z = \frac{-2x_1 + 2x_2 + 9}{x_1 + 4x_2 + 22}$$

$$17. \quad \begin{cases} 7x_1 - 4x_2 \geq -48, \\ 4x_1 + 5x_2 \geq 40, \\ 7x_1 - 5x_2 \leq 15, \\ 4x_1 + 3x_2 \leq 73, \end{cases}$$

$$\max Z = \frac{-6x_1 + 2x_2 + 3}{3x_1 + 7x_2 + 2}$$

$$18. \quad \begin{cases} 2x_1 - 3x_2 \geq -20, \\ 4x_1 + 3x_2 \geq 32, \\ 2x_1 - 5x_2 \leq 24, \\ 4x_1 + 7x_2 \leq 116, \end{cases}$$

$$\max Z = \frac{-2x_2 + 3}{3x_1 + x_2 + 12}$$

$$19. \quad \begin{cases} -\frac{1}{3}x_1 + 3x_2 \geq -1, \\ -x_1 + \frac{5}{2}x_2 \leq \frac{5}{2}, \\ 7x_1 + 11x_2 \leq 77, \\ x_1 \geq 3, \end{cases}$$

$$\min Z = \frac{4x_1 + 3x_2 + 13}{-3x_1 + 9x_2 + 2}$$

$$20. \quad \begin{cases} \frac{1}{2}x_1 + 3x_2 \geq \frac{3}{2}, \\ -\frac{1}{2}x_1 + 3x_2 \geq -\frac{3}{2}, \\ 3x_1 + 10x_2 \leq 30, \\ -3x_1 + 9x_2 \leq 27, \end{cases}$$

$$\min Z = \frac{-3x_1 + 2x_2}{x_1 + 5x_2 + 5}$$

$$21. \quad \begin{cases} 4x_1 - 3x_2 \geq -16, \\ 4x_1 + 5x_2 \geq 48, \\ 2x_1 - 3x_2 \leq 32, \\ 6x_1 + 7x_2 \leq 160, \end{cases}$$

$$\max Z = \frac{2x_1 + 4x_2 + 9}{7x_2 + 12}$$

$$22. \quad \begin{cases} 2x_1 + 10x_2 \geq 20, \\ x_1 - 3x_2 \geq -12, \\ 3x_1 + \frac{5}{2}x_2 \leq \frac{27}{2}, \\ x_1 \leq 2, \end{cases}$$

$$\max Z = \frac{6x_1 + 3}{8x_1 + 3x_2 + 1}$$

$$23. \quad \begin{cases} 4x_1 - 3x_2 \geq -24, \\ x_1 + 2x_2 \geq 12, \\ 3x_1 - 4x_2 \geq 16, \\ 4x_1 + 5x_2 \leq 104, \end{cases}$$

$$\min Z = \frac{-9x_1 - 2x_2 - 5}{x_1 + 4x_2 + 11}$$

$$24. \quad \begin{cases} 5x_1 - 3x_2 \geq -30, \\ 3x_1 + 4x_2 \geq 32, \\ 2x_1 - 5x_2 \leq 6, \\ 7x_1 + 6x_2 \leq 162, \end{cases}$$

$$\min Z = \frac{5x_1 - 7x_2 + 4}{-3x_1 + 9x_2 + 3}$$

$$25. \quad \begin{cases} x_1 + 5x_2 \geq 5, \\ x_1 + x_2 \leq 7, \\ -4x_1 + 5x_2 \geq -20, \\ x_1 - 2x_2 \geq -2, \end{cases}$$

$$\max Z = \frac{-5x_1 + 5x_2 + 5}{-7 + 7x_2 + 7}$$

$$26. \quad \begin{cases} 3x_1 - 4x_2 \geq -40, \\ 3x_1 + 7x_2 \geq 56, \\ 3x_1 - 3x_2 \leq 22, \\ 5x_1 + 6x_2 \leq 136, \end{cases}$$

$$\min Z = \frac{6x_1 + 3x_2 + 24}{x_1 - 5x_2 + 12}$$

$$27. \quad \begin{cases} x_1 - 2x_2 \geq -10, \\ 3x_1 + 5x_2 \geq 36, \\ x_1 - 3x_2 \leq 14, \\ 4x_1 + 5x_2 \leq 90, \end{cases}$$

$$\min Z = \frac{-6x_1 + 12x_2 + 44}{x_1 + 8x_2 + 24}$$

$$28. \quad \begin{cases} \frac{5}{3}x_1 - x_2 \geq -10, \\ x_1 + \frac{4}{3}x_2 \geq \frac{32}{3}, \\ x_1 - \frac{5}{2}x_2 \leq 3, \\ 7x_1 + 6x_2 \leq 162, \end{cases}$$

$$\max Z = \frac{2x_1 + 3x_2 + 24}{3x_1 + 17x_2 + 52}$$

$$29. \quad \begin{cases} 7x_1 - 8x_2 \geq -12, \\ 5x_1 + 2x_2 \geq 30, \\ 3x_1 - 8x_2 \leq 48, \\ 3x_1 + 4x_2 \leq 84, \end{cases}$$

$$\max Z = \frac{-2x_1 + 7x_2 + 13}{3x_1 + x_2 + 24}$$

$$30. \quad \begin{cases} 5x_1 - 4x_2 \geq -14, \\ x_1 + x_2 \geq 8, \\ 2x_1 - 5x_2 \leq 32, \\ 3x_1 + 4x_2 \leq 94, \end{cases}$$

$$\min Z = \frac{-x_1 + 5x_2 + 13}{-3x_1 + 4x_2 + 22}$$

Практичне заняття № 8

Наближені методи оптимізації. Прямі методи безумовної оптимізації (методи нульового порядку): покоординатного спуску, пошук за деформованим багатогранником

Мета: вивчення понять релаксаційної послідовності для заданої функції, мінімізаційної послідовності, швидкості збіжності методу, критеріїв оптимальності для задач безумовної і умовної оптимізації; ідеї та алгоритмів прямих методів безумовної оптимізації

План заняття

1. Постановка задачі багатовимірної оптимізації
2. Класифікація методів оптимізації. Поняття збіжності
3. Оцінка швидкості збіжності. Критерії закінчення ітераційного процесу
4. Прямі методи безумовної оптимізації (методи нульового порядку): метод покоординатного спуску – алгоритм Гаусса, пошук за деформованим багатогранником

Опорні поняття

1. Постановка задачі багатовимірної оптимізації

Задачу багатовимірної *умовної оптимізації* записують у такій спосіб:

$$f(x) \rightarrow \min_{x \in \Omega} \quad (8.1)$$

$$\Omega = \{x \in D: g_i(x) \leq 0, \quad 1 \leq i \leq m; \quad g_i(x) = 0, \quad m + 1 \leq i \leq s\}$$
$$D = \{x \in R^n: a_j \leq x_j \leq b_j, \quad 1 \leq j \leq n\},$$

де $f: R^n \rightarrow R, g_i: R^n \rightarrow R, 1 \leq i \leq s$ – задані функції, $n \geq 2$ – розмірність задачі, D – область пошуку.

Задача багатовимірної *безумовної оптимізації* має такий вигляд:

$$f(x) \rightarrow \min_{x \in R^n} \quad (8.2)$$

Якщо функція багатьох змінних $f(x) = f(x_1, x_2, \dots, x_n)$ є двічі диференційовною, то задачу (8.2) можна розв'язати аналітично, застосовуючи критерій безумовного екстремуму.

В цьому випадку необхідно записати необхідні умови оптимальності, а саме:

$$f'(x) = 0 \quad \text{або} \quad \frac{\partial f(x)}{\partial x_i} = 0, \quad i = 1, \dots, n, \quad (8.3)$$

і знайти всі стаціонарні точки функції $f(x)$. Потім серед них, використовуючи достатні умови, відбираємо точки локального мінімуму, в яких матриця других похідних $f''(x)$ є додатно визначеною, і, порівнюючи значення цільової функції в цих точках, знаходимо точку глобального мінімуму.

Однак аналітично розв'язати систему рівнянь (8.3) не завжди можливо. Крім того, функція $f(x)$ може бути недиференційовною або не заданою аналітично. Тому описаний спосіб розв'язання задачі (8.2) має обмежене коло застосувань і на практиці частіше застосовуються наближені обчислювальні методи.

2. Класифікація методів оптимізації. Поняття збіжності

Будь-який обчислювальний метод розв'язання задачі (8.1) або (8.2) заснований на обчисленні значень цільової функції, а також (часто) її похідних. Метод називається *пасивним*, якщо точки, в яких здійснюються обчислення, вибираються незалежно одна від одної і в принципі визначені заздалегідь, та *послідовним*, якщо такі точки вибираються послідовно в процесі розрахунку на основі інформації, отриманої протягом обчислювального процесу. Зрозуміло, послідовних методів значно більше.

Послідовний метод генерує послідовність точок $x^{(1)}, x^{(2)}, \dots, x^{(k)} \dots$ в R^n , які називаються *наближеннями* до розв'язку задачі. Ця послідовність $\{x^{(k)}\}$, котру називають *траєкторією методу*, може містити лише частину множини точок, в яких здійснюються обчислення. Перехід від $x^{(k)}$ до наступного наближення $x^{(k+1)}$ називається *кроком* або *ітерацією* методу. Спосіб цього переходу обумовлює сутність методу, і тому часто методи записують у вигляді ітераційної формули, тобто

$$x^{(k+1)} = \Psi_k(x^{(k)}), k = 0, 1, 2, \dots, \quad (8.4)$$

де Ψ_k – відображення із значеннями в \mathbb{R}^n , визначене на деякій множині $U_k \subset \mathbb{R}^n$, причому передбачається, що до початку $(k + 1)$ -го кроку воно є відомим.

Якщо ітераційна схема фіксована, то вибір початкового наближення $x^{(0)} \in U_0$ коректним чином визначає траєкторію методу тоді і тільки тоді, коли

$$\Psi_{k-1} \left(\Psi_{k-2} \left(\dots \Psi_0(x^{(0)}) \dots \right) \right) \in U_k \quad \forall k = 1, 2, \dots$$

Розглянута ітераційна схема (8.4) називається *однокроковою*. Іноді застосовують багатокрокові схеми, коли Ψ_k залежить не тільки від $x^{(k)}$, але й від деяких попередніх наближень розв'язку.

Якщо для точок $x^{(1)}, x^{(2)}, \dots, x^{(k)} \dots$ виконується нерівність

$$f(x^{(0)}) \geq f(x^{(1)}) \geq \dots \geq f(x^{(k)}) \geq \dots, \quad (8.5)$$

то таку послідовність точок $\{x^{(k)}\}$ називають *мінімізаційною* для функції $f(x)$.

Коли ж має місце співвідношення

$$f(x^{(0)}) > f(x^{(1)}) > \dots > f(x^{(k)}) > \dots, \quad (8.6)$$

то $\{x^{(k)}\}$ називають *релаксаційною послідовністю* для $f(x)$.

Визначення 8.1. Вектор $d \in R^n$ називається напрямком спадання функції $f: R^n \rightarrow R$ в точці $x \in R^n$, якщо для будь-якого достатньо малого $t > 0$ є справедливою нерівність $f(x + td) < f(x)$.

Множина всіх напрямків спадання функції f в точці $x \in R^n$ є конусом і позначається $D_f(x)$. Відтак, $d \in D_f(x)$ у тому і тільки тому випадку, коли будь-яке достатньо мале зрушення з точки x в напрямку d призводить до зменшення функції. Буде виконуватись наступна лема.

Лема 8.1. Нехай функція $f: R^n \rightarrow R$ є диференційовною в точці $x \in R^n$. Тоді: 1) для будь-якого $d \in D_f(x)$ виконується $(f'(x), d) \leq 0$; 2), якщо $d \in R^n$ задовольняє умову $(f'(x), d) < 0$, то $d \in D_f(x)$.

Загальна ітераційна формула релаксаційних методів (*методів спуску*) для задачі (8.2) має такий вигляд:

$$x^{(k+1)} = x^{(k)} + \alpha_k d^{(k)}, k = 0, 1, 2, \dots, \quad (8.7)$$

де $d^{(k)}$ – вектор, що задає напрямок спуску, $\alpha_k > 0$ – кроковий множник.

Зауважимо, що α_k вибирається, аби виконувалась принаймні нерівність (8.6), тобто послідовність $\{f(x^{(k)})\}$ має бути монотонно спадною.

Передбачається, що $D_f(x) \neq \emptyset$; якщо $D_f(x) = \emptyset$, то процес завершують. Варто помітити, оскільки $d^{(k)} \in D_f(x)$, число $\alpha_k > 0$, яке забезпечує виконання умови (8.6), завжди існує. Отже, здійснюється спуск генерованих наближень на поверхні все більш низького рівня цільової функції задачі, що дозволяє сподіватися на збіжність послідовності таких наближень до розв'язку.

Конкретний метод спуску характеризується способом вибору напрямку спадання, а також використовуваною процедурою вибору параметра довжини кроку. Такі процедури, як правило, засновані на дослідженні звуження функції f на промінь, який виходить з точки $x^{(k)}$ в напрямку вектора $d^{(k)}$, тому їх називають *процедурами одновимірного пошуку*.

З леми 8.1 випливає, що коли $f'(x) \neq 0$, то як напрямок спуску можна задати антиградієнт: $d^{(k)} = -f'(x^{(k)})$.

Порядком методу є максимальний порядок похідних цільової і обмежувальних функцій задачі (8.1) або (8.2), які використовуються для здійснення однієї ітерації методу. Так, якщо під час вибору вектора p_k застосовується інформація лише про значення функції, то мають справу з методами нульового порядку. Якщо ж залучаються перші (другі і т.д.) похідні, то схема (8.7) є ітераційною формулою методів першого (другого і т.д.) порядку. Спосіб вибору крокового множника визначає модифікацію того чи іншого методу.

Розглянемо основні процедури одновимірного пошуку, передбачаючи, що для поточного наближення $x^{(k)}$ вже отриманий напрямок $d^{(k)} \in D_f(x)$.

Правило одновимірної мінімізації. Параметр $\alpha_k > 0$ вибирається з умови

$$f(x^{(k)} + \alpha_k d^{(k)}) = \min_{\alpha \geq 0} f(x^{(k)} + \alpha d^{(k)}),$$

тобто як розв'язок одновимірної задачі оптимізації

$$\varphi_k(\alpha) \rightarrow \min, \quad \alpha \in R_+, \quad (8.8)$$

де

$$\varphi_k: R_+ \rightarrow R, \quad \varphi_k(\alpha) = f(x^{(k)} + \alpha d^{(k)}).$$

Зауважимо, що при цьому, якщо функція f є диференційовною в точці $x^{(k+1)}$, то буде правильною наступна рівність:

$$0 = \varphi'_k(\alpha) = (f'(x^{(k)} + \alpha d^{(k)}), d^{(k)}) = (f'(x^{(k+1)}), d^{(k)}). \quad (8.9)$$

Тобто якщо $f'(x^{(k+1)}) \neq 0$, то геометрично $x^{(k+1)}$ є точкою дотику променю, що виходить з $x^{(k)}$ в напрямку $d^{(k)}$, і поверхні рівня функції f , яка проходить через точку $x^{(k+1)}$.

Такий спосіб вибору параметра α_k є найкращим у тому сенсі, що дозволяє отримати наступне наближення з найменшим значенням цільової функції f вздовж використовуваного напрямку спадання. Якщо f – квадратична функція, тобто

$$f(x) = (Ax, x) + (b, x), \quad x \in R^n,$$

де $A \in R(n, n)$ – симетрична матриця, $b \in R^n$, причому матриця A є додатно визначеною, то розв'язок α_k задачі (8.8) обчислюється за явною формулою:

$$\alpha_k = -\frac{(2Ax^{(k)} + b, d^{(k)})}{2(Ad^{(k)}, d^{(k)})}.$$

Однак в загальному випадку цей спосіб визначення α_k є достатньо трудомістким.

Для розв'язання на кожному кроці одновимірної оптимізації (8.8) можуть бути застосовані методи, розглянуті вище.

Крім того, можна застосовувати менш витратні способи вибору параметра довжини кроку, які будуть описані нижче.

Іноді (а на практиці майже завжди, за виключенням спеціальних випадків, наприклад, квадратичної цільової функції) задачу (8.8) замінюють задачею

$$\varphi_k(\alpha) \rightarrow \min, \quad \alpha \in [0, \bar{\alpha}],$$

де $\bar{\alpha} > 0$ – фіксований параметр.

У подальшому, говорячи про правило одновимірної оптимізації, будемо мати на увазі саме цю допоміжну задачу, вважаючи, що вибір $\bar{\alpha} = +\infty$ відповідає задачі (8.8).

Правило Арміхо. Цей спосіб передбачає диференційовність функції f в поточній точці $x^{(k)}$.

Фіксуємо числа $\bar{\alpha} > 0, \varepsilon, \theta \in (0,1)$. Нехай $\alpha = \bar{\alpha}$.

1. Перевіряємо виконання умови

$$f(x^{(k)} + \alpha d^{(k)}) \leq f(x^{(k)}) + \varepsilon \alpha (f'(x^{(k)}), d^{(k)}). \quad (8.10)$$

2. Якщо (8.10) не виконується, то замінюємо α на $\theta \alpha$ і переходимо до п.1. У протилежному випадку будемо вважати, що $\alpha_k = \alpha$.

Отже, α_k обчислюється як перше з чисел α , отриманих в результаті дроблення початкового значення $\bar{\alpha}$, для якого виконується нерівність (8.10). Наступна лема показує, що у випадку, коли $d^{(k)}$ задовольняє достатню умову напрямку спадання, сформульовану в лемі 1, тобто

$$(f'(x^{(k)}), d^{(k)}) < 0, \quad (8.11)$$

то кількість дроблень буде скінченною.

Лема 8.2. Нехай функція $f: R^n \rightarrow R$ є диференційовною в точці $x^{(k)} \in R^n$. Тоді якщо $d^{(k)} \in R^n$ задовольняє умову (8.11), то нерівність (8.10) має місце для будь-якого достатньо малого $\alpha > 0$.

Вочевидь, якщо виконується (8.11), вибір α_k за правилом Арміхо гарантує виконання умови монотонного спадання (8.6). Більш того, нерівність (8.10) при $\alpha = \alpha_k$ демонструє оцінку того, наскільки $f(x^{(k+1)})$ менше за $f(x^{(k)})$, і вона зазвичай виявляється достатньою для обґрунтування збіжності методу, на відміну від умови (8.6).

Лема 8.3. Нехай функція $f: R^n \rightarrow R$ є диференційовною на R^n і її похідна є неперервною за Ліпшицем на R^n з константою Ліпшиця $L > 0$. Тоді, якщо для деяких $x^{(k)}, d^{(k)} \in R^n$ виконується умова (8.11), нерівність (8.10) має місце для будь-якого $\alpha \in (0, \bar{\alpha}_k]$, де

$$\bar{\alpha}_k = - \frac{2(1 - \varepsilon)(f'(x^{(k)}), d^{(k)})}{L \|d^{(k)}\|^2} > 0. \quad (8.12)$$

Правило постійного параметра. Фіксуємо (незалежне від k) число $\bar{\alpha} > 0$ і вважаємо, що $\alpha_k = \bar{\alpha}$. Це правило просте, його застосовують у тих випадках, коли обчислення значень цільової функції задачі є трудомісткою операцією.

Теоретичний аналіз методів з достатньо малим постійним параметром довжини кроку зводиться до аналізу методів, що використовують правило Арміхо.

Якщо вдається визначити константу Ліпшиця або оцінити її зверху, то формула (8.12) може бути використана для явного обчислення параметрів довжини кроку.

Розглянуті три правила одновимірного пошуку вважаються, зазвичай, основними, але на практиці часто користуються і іншими. Одним з них є *правило Голдстейна* – вибір параметра довжини кроку, який задовольняє подвійній нерівності

$$\varepsilon_1 \leq \frac{f(x^{(k)} + \alpha d^{(k)}) - f(x^{(k)})}{\alpha(f'(x^{(k)}), d^{(k)})} \leq \varepsilon_2 \quad (8.13)$$

при фіксованих $\varepsilon_1, \varepsilon_2 \in (0,1), \varepsilon_1 < \varepsilon_2$.

Тут ліва частина є нерівністю Арміхо, коли $\varepsilon = \varepsilon_1$; вона забезпечує спадання цільової функції. Разом з тим, згідно з лемою 8.2, нерівність Арміхо виконується для будь-якого достатньо малого $\alpha > 0$, на відміну від правої частини (8.13), яка порушується для всіх α , які є достатньо близькими до нуля. В цьому і полягає сенс введення правої нерівності: вона не дозволяє вибирати надто малі параметри довжини кроку, тим самим перешкоджаючи уповільненню методу.

Іншою реалізацією тієї самої ідеї є *правило Вулфа*, яке формулюється так само як попереднє, але замість (8.13) використовуються такі нерівності:

$$f(x^{(k)} + \alpha d^{(k)}) \leq f(x^{(k)}) + \varepsilon_1 \alpha(f'(x^{(k)}), d^{(k)}), \quad (8.14)$$

$$(f'(x^{(k)} + \alpha d^{(k)}), d^{(k)}) \geq \varepsilon_2 (f'(x^{(k)}), d^{(k)}). \quad (8.15)$$

У тих випадках, коли обчислення похідної функції f не є трудомістким, правило Вулфа визнається найбільш ефективним відомим правилом одновимірного пошуку. Важлива властивість цього правила пов'язана з квазіньютонівськими методами (див. Практичне заняття № 12).

Наведемо процедуру, яка реалізує правило Вулфа.

Фіксуємо числа $\varepsilon_1, \varepsilon_2 \in (0,1), \varepsilon_1 < \varepsilon_2$. Нехай $\hat{\alpha} = \check{\alpha} = 0$. Вибираємо початкове значення $\alpha > 0$.

1. Перевіряємо виконання нерівностей (8.14) і (8.15). Якщо обидві вони виконуються, то переходимо до п. 6.

2. Якщо порушена нерівність (8.14), то вважаємо, що $\hat{\alpha} = \alpha$, і переходимо до п. 5.

3. Якщо порушена нерівність (8.15), то вважаємо, що $\check{\alpha} = \alpha$.

4. Якщо $\hat{\alpha} = 0$, то вибираємо нове пробне значення $\alpha > \check{\alpha}$ («екстраполяція») і переходимо до п.1.

5. Вибираємо нове пробне значення $\alpha \in (\check{\alpha}, \hat{\alpha})$ («інтерполяція») і переходимо до п.1.

6. Вважаємо, що $\alpha_k = \alpha$.

Порушення нерівності (8.14) означає, що поточне пробне значення α надто велике, а порушення (8.15) надто мале. Описана процедура працює таким чином. Спочатку реалізуються кроки «екстраполяції», допоки $\hat{\alpha}$ не стане додатним. Після цього виконуються кроки «інтерполяції»; при цьому $\hat{\alpha}$ може

тільки зменшуватися, залишаючись додатним, а $\check{\alpha}$ – лише збільшуватися, завжди залишаючись меншим за $\hat{\alpha}$.

Екстраполяція і інтерполяція в наведеній процедурі запроваджуються багатьма способами. Наприклад, можна зафіксувати числа $\theta_1 > 1$, $\theta_2 \in (0,1)$ і під час екстраполяції замінити α на $\theta_1\alpha$, а для здійснення інтерполяції встановити $\alpha = (1 - \theta_2)\check{\alpha} + \theta_2\hat{\alpha}$. Важливо, аби у випадку нескінченної кількості кроків екстраполяції величина $\check{\alpha}$ необмежено зростала, а у випадку нескінченного числа кроків інтерполяції величина $\hat{\alpha} - \check{\alpha}$ прямувала до нуля.

До обчислювальних методів безумовної оптимізації функції багатьох змінних відносять, наприклад, методи нульового порядку – конфігурацій (метод Хука – Дживса), пошук за деформованим багатогранником (метод Нелдера – Міда), методи першого порядку – градієнтні, спряжених градієнтів, метод другого порядку – метод Ньютона та ін.

Якщо будь-яке початкове наближення $x^{(0)} \in U_0$ коректним чином визначає траєкторію методу, деяка точка якої збігається з шуканим розв'язком задачі, то метод називається *скінченнокроковим* або *скінченним*. Такі методи розроблені для задач лінійного і квадратичного програмування. Для більш загальних задач оптимізації зазвичай мають справу з *нескінченнокроковими* методами, траєкторії яких, взагалі кажучи, не проходять через точний розв'язок, а лише апроксимують його в деякому сенсі. Характер такої апроксимації визначає *збіжність* методу.

Наприклад, якщо

$$\{x^{(k)}\} \rightarrow x^* \quad (k \rightarrow \infty),$$

де x^* – шуканий розв'язок задачі, то кажуть про *збіжність за аргументом*, або просто про збіжність методу. Іноді вдається встановити не збіжність методу до конкретного розв'язку, а лише більш слабку властивість:

$$\text{dist}(x^{(k)}, X_*) \rightarrow 0 \quad (k \rightarrow \infty),$$

де X_* – множина розв'язків задачі (8.2). В цьому разі говорять про *збіжність до множини розв'язків*. Якщо

$$f(x^{(k)}) \rightarrow f^* \quad (k \rightarrow \infty),$$

де f^* – мінімальне значення цільової функції, то кажуть про *збіжність за функцією*, а саму послідовність називають мінімізуючою. Безумовно, сама мінімізуюча послідовність може й не збігатися. Наприклад, для функції $f(x) = \frac{x^2}{1+x^4}$ мінімальне значення $f^* = 0$, точка мінімуму $x^* = 0$, послідовність точок $x^{(k)} = k$ є мінімізуючою, оскільки виконується збіжність $f(x^{(k)}) \rightarrow f^*$, хоча сама послідовність точок $x^{(k)}$ є розбіжною.

Часто вдається показати збіжність методу не до множини (локальних чи глобальних) розв'язків, а лише до множини стаціонарних точок задачі або ж прямування до нуля відхилу відповідних необхідних умов оптимальності. Наприклад, у випадку задачі безумовної оптимізації (8.2) виконання

$$\{f'(x^{(k)})\} \rightarrow 0 \quad (k \rightarrow \infty)$$

визначає збіжність методу за градієнтом.

Нарешті, якщо $U_0 = R^n$ і має місце збіжність методу з будь-якого початкового наближення $x^0 \in R^n$, то кажуть про *глобальну збіжність*. У випадку, коли той чи інший тип збіжності вдається гарантувати лише з початкових наближень, достатньо близьких до шуканого розв'язку (множини розв'язків, стаціонарній точці, множині стаціонарних точок), то кажуть про *локальну збіжність* методу.

В багатьох випадках метод розв'язання складної задачі є редукцією до більш простих або послідовності таких задач. Зрозуміло, це має сенс, коли для простих задач відомі ефективні методи розв'язання. Наприклад, класичний метод Ньютона зводить розв'язання системи нелінійних рівнянь до послідовного розв'язання лінійних систем, для яких розвинені ефективні методи обчислювальної лінійної алгебри. Аналогічно, методи розв'язання загальних задач умовної оптимізації часто використовують допоміжні процедури методів одновимірної оптимізації, безумовної оптимізації, лінійного чи квадратичного програмування. Методи безумовної оптимізації у свою чергу залучають методи одновимірної оптимізації, а методи квадратичного програмування використовують методи лінійного програмування. Тому, природно, що розвиток методів, які знаходяться вище за ієрархією, неможливий без розвитку методів, нижчих за неї.

3. Оцінка швидкості збіжності. Критерії закінчення ітераційного процесу

Однією з основних характеристик ефективності обчислювального методу оптимізації є швидкість його збіжності. Будемо далі мати на увазі швидкість збіжності за аргументом; термінологія для інших видів швидкості є аналогічною.

Нехай метод локально збігається до точки мінімуму x^* . Оцінки швидкості збіжності виражають гарантовану швидкість спадання величини $\|x^{(k)} - x^*\|$ коли $k \rightarrow \infty$ для збіжних до x^* траєкторій $\{x^{(k)}\}$. Тут всюди передбачається, що початкове наближення $x^0 \in R^n$ є достатньо близьким до x^* , траєкторії, що розглядаються, не влучають в x^* на жодному кроці, а всі виникаючі нижче константи не залежать від конкретного вибору x^0 . Якщо

$$\limsup_{k \rightarrow \infty} \frac{\|x^{(k+1)} - x^*\|}{\|x^{(k)} - x^*\|} = q,$$

де $q \in (0,1)$ — деяка константа, то говорять, що метод збігається з лінійною швидкістю, а якщо $q = 0$, то з надлінійною. Швидкість збіжності, що є більш низькою, ніж лінійна, називається сублінійною. Важливим окремим випадком надлінійної є квадратична швидкість збіжності, коли

$$\limsup_{k \rightarrow \infty} \frac{\|x^{(k+1)} - x^*\|}{\|x^{(k)} - x^*\|^2} \leq C,$$

тут і далі $C > 0$ – деяка константа.

Прикладом оцінки, яка гарантує лише сублінійну швидкість збіжності, слугує така нерівність:

$$\limsup_{k \rightarrow \infty} k \|x^{(k)} - x^*\| \leq C.$$

У випадку її виконання говорять про арифметичну швидкість збіжності методу. Якщо ж

$$\limsup_{k \rightarrow \infty} \frac{\|x^{(k)} - x^*\|}{q^k} \leq C$$

за деякого $q \in (0,1)$, то кажуть про геометричну швидкість збіжності. Вочевидь, лінійна швидкість збіжності передбачає геометричну, але, взагалі кажучи, не навпаки.

Оцінки швидкості збіжності дають важливий спосіб якісного порівняння різних методів один з одним. Разом з тим вона не є єдиною характеристикою ефективності методу. Необхідно, приміром, брати до уваги трудомісткість однієї ітерації, причому зазвичай методи з більш високою швидкістю збіжності мають більш трудомістку ітерацію. Метод, який швидко збігається, з дорогою ітерацією може програвати більш повільному методу, кожна ітерація якого є дешевшою. Друга важлива характеристика – стійкість методу по відношенню до впливу збурень вхідних даних і перешкод, які завжди існують у реальних обчислювальних процесах.

Вибираючи метод, варто брати до уваги такі міркування. Якщо потрібно розв'язати одну конкретну задачу оптимізації, то перевагу віддають надійному методу, з не найвищою швидкістю збіжності. Якщо ж мова йде про серію однотипних задач, які, наприклад, відрізняються значеннями деяких параметрів, то швидкість збіжності виходить на перший план. В останньому випадку доцільно витратити час на вибір і налаштування придатного швидкого алгоритму.

Крім того, на практиці обчислювальний процес не може здійснюватися нескінченно, тому будь-який нескінченнокроковий метод має супроводжуватися правилом зупинки, і найбільш надійні правила такого ґтибу отримують за допомогою оцінок збіжності. Дійсно, наявність подібної оцінки дозволяє вказати кількість кроків методу, які гарантують досягнення заданої точності (за аргументом або в деякому іншому сенсі).

На практиці зазвичай використовують менш надійні правила зупинки за непрямыми ознаками, приміром, за поведінкою частини траєкторії $\{x^{(k)}\}$, отриманої до поточного кроку, або послідовності $\{f(x^{(k)})\}$, а також за величиною відхилу необхідних умов оптимальності. Наприклад, фіксують значення $\varepsilon > 0$ і зупиняють метод після $(k + 1)$ -ї ітерації, якщо

$$\|x^{(k+1)} - x^{(k)}\| \leq \varepsilon$$

або

$$|f(x^{(k+1)}) - f(x^{(k)})| \leq \varepsilon.$$

А у випадку задачі безумовної оптимізації використовують також нерівність

$$\|f'(x^{(k)})\| \leq \varepsilon.$$

Ясно, що такі правила зупинки не є достатньо надійними і не гарантують близькість отриманого наближення до шуканого розв'язку. Приміром, із граничного співвідношення $\lim_{k \rightarrow \infty} \|x^{(k+1)} - x^{(k)}\| = 0$ не випливає збіжність послідовності $\{x^{(k)}\}$, що видно на прикладі послідовності $\{x^{(k)}\} \subset R$, $x^{(k)} = \sqrt{k}$ або $x^{(k)} = \sin\sqrt{k}$, $k = 0, 1, 2, \dots$. Проте вони використовуються часто, оскільки інших конструктивних правил зупинки немає.

Крім того, кількість кроків методу може (і повинна) визначатися наявними обчислювальними ресурсами. Часто, якщо процес не зупиняється раніше за одним з вище зазначених критеріїв, то його доводиться зупинити з причини вичерпання ресурсів. Зазвичай задають максимальний час розрахунків, максимальну кількість ітерацій, максимальну кількість обчислень значень цільової функції і т.і. Зрозуміло, що отримане у такому випадку наближення може не мати до шуканого розв'язку ніякого відношення.

Отже, найбільш правильним є комбінування названих правил зупинки, що передбачає задання деякої їхньої ієрархії.

4. Прямі методи (методи нульового порядку): метод покоординатного спуску – алгоритм Гаусса, пошук за деформованим багатогранником

Прямі методи або методи нульового порядку не вимагають знання цільової функції у явному вигляді. Вони не вимагають регулярності та неперервності цільової функції, а також існування її похідних, що є істотною перевагою при розв'язанні складних технічних та економічних задач.

Здебільшого ці методи носять евристичний характер і призначені для розв'язання безумовних задач оптимізації.

Метод покоординатного спуску

Траєкторія пошуку екстремуму функції $f(x)$ за даним методом вибирається у вигляді ламаної лінії, відрізки якої є паралельними координатним осям простору змінних $x \in E^n$.

У випадку $n = 2$ можна на прикладі ліній рівня $f(x) = \text{const}$ показати дану траєкторію.

Опишемо ідею методу. Після вибору деякого початкового наближення $x^0 = (x_1^0, x_2^0, \dots, x_n^0)$ шукаємо $\min_{x_1} f(x) = f(x)$ за фіксованих значень

$x_2^0, x_3^0, \dots, x_n^0$. Таким чином, рух з точки x^0 здійснюється по прямій, паралельній осі Ox_1 в напрямку спадання функції $f(x)$, $\min_{x_1} f(x) = x_1^{(1)}$. Далі пошук йде вздовж осі Ox_2 з точки з координатами $(x_1^{(1)}, x_2^0, \dots, x_n^0)$, тобто шукаємо $\min_{x_2} f(x)$. Описана процедура послідовно повторюється для всіх $x_i, i = 1, \dots, n$.

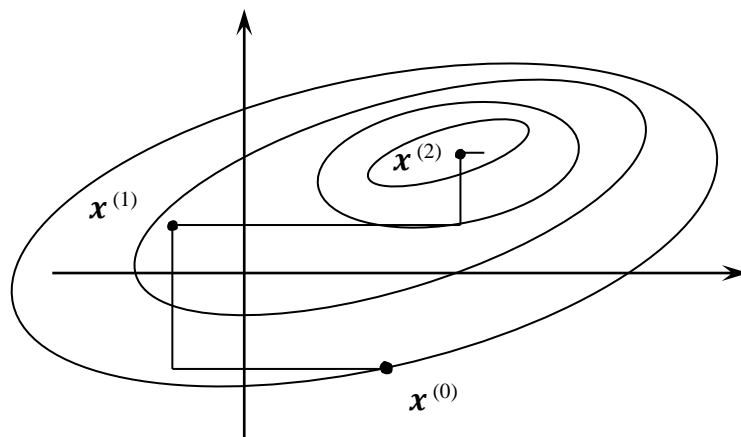


Рис. 8.1. Траєкторія руху точки за методом покоординатного спуску

Після завершення пошуку за всіма x_i отримуємо точку $x^1 = (x_1^1, x_2^1, \dots, x_n^1)$ (рис. 8.1). Процес повторюється аналогічно викладеному, і як результат отримуємо точку $x^2 = (x_1^2, x_2^2, \dots, x_n^2)$ і так далі.

Ітераційний процес пошуку закінчується, коли зміна аргументу x мало впливає на зміну функції $f(x)$, тобто виконується нерівність:

$$|f(x^{k+1}) - f(x^k)| < \varepsilon \quad (8.16)$$

де ε – задана точність обчислень.

Зауваження. Під час пошуку мінімуму функції $f(X)$ в напрямку вздовж осі $Ox_i, i = 1, \dots, n$ перехід від поточної точки x_i^j до точки x_i^{j+1} здійснюється відповідно до такої формули:

$$x_i^{j+1} = x_i^j + \text{sign} \cdot \alpha, \quad (8.17)$$

де параметр sign може приймати значення 1 або -1 , а α – величина кроку в даному напрямку. Для визначення $\min_{x_i} f(x)$ коефіцієнт α змінюється від початкового значення α_0 до мінімально можливого α_{\min} .

Алгоритм методу покоординатного спуску

1. Задаємо точність обчислення ε , початковий крок α_0 , мінімально допустимий крок α_{\min} , точку початкового наближення $x^0 \in E^n$, порядковий номер ітерації $k = 0$ і обчислюємо $f(x^k)$.

2. Запам'ятовуємо $f(x^k); k = k + 1, i = 0$.
3. Починаємо пошук паралельно осі $Ox_i: i = i + 1$, якщо $i > n$, переходимо до п. 11, інакше – до п. 4.
4. Задаємо крок $\alpha = \alpha_0$.
5. Задаємо знак $sign = 1$.
6. Обчислюємо x_i^{j+1} за формулою (8.17) і значення функції в новій точці.
7. Якщо нове значення функції є меншим за попереднє, то x_i^j вважаємо як x_i^{j+1} , $f(x^j)$ запам'ятовуємо як $f(x^{j+1})$ і повертаємося до п. 6, інакше змінюємо знак $sign$ на протилежний.
8. Якщо знак $sign < 0$, то переходимо до п. 6, інакше – до п. 9.
9. Зменшуємо величину кроку α , ($\alpha^{j+1} = 0.1 \cdot \alpha^j$).
10. Якщо поточний крок $\alpha < \alpha_{min}$, то переходимо до п. 3, інакше – до п. 5.
11. Перевіряємо умову досягнення заданої точності:

$$|f(x^{(k+1)}) - f(x^{(k)})| < \varepsilon,$$

і у випадку її виконання переходимо до п. 12, інакше – до п. 2.

12. Завершуємо обчислення, прийнявши за точку мінімуму x^* останнє значення x^k , $\min f(x) = f(x^*)$.

Метод є простим, але не дуже ефективним. Проблеми можуть виникнути, наприклад, коли лінії рівня цільової функції навіть двох змінних є сильно витягнутими і «еліпсоїди» орієнтовані вздовж прямих вигляду $x_1 = x_2$. В подібній ситуації пошук швидко застряє на дні яру, а коли початкове наближення виявляється на осі «еліпсоїда», то процес так і залишиться в цій точці. Гарні результати отримують у тих випадках, коли цільова функція є *сепарабельною*, тобто $f(x) = \sum_{i=1}^n f_i(x_i)$.

Метод Нелдера – Міда (пошук за деформованим багатогранником)

Це метод безумовної оптимізації функції декількох змінних, який не використовує похідні функції, а тому легко може бути застосований до негладких або зашумлених функцій. Ідея методу полягає в послідовному переміщенні і деформації симплекса навколо точки екстремуму. Однак техніка Нелдера – Міда – це евристичний метод пошуку, який може збігатися до нестационарних точок відносно проблем, які можуть бути вирішені альтернативними методами. Методика Нелдера – Міда була запропонована Джоном Нелдером та Роджером Мідом у 1965 р. як розвиток методу Спендлі та ін.

Припустимо потрібно знайти безумовний мінімум функції n змінних $f(x^{(1)}, x^{(2)}, \dots, x^{(n)})$. Передбачається, що серйозних обмежень на область визначення функції немає, тобто функція визначена в усіх точках, які зустрічаються.

Параметрами методу є:

- коефіцієнт відображення $\alpha > 0$, зазвичай дорівнює 1.
- коефіцієнт стиснення $\beta > 0$, часто його обирають таким, що дорівнює 0.5.
- коефіцієнт розтягування $\gamma > 0$, зазвичай дорівнює 2.

Кроки алгоритму наступні:

1. **«Підготовка»**. Спочатку обирається $n + 1$ точка $x_i = (x_i^{(1)}, x_i^{(2)}, \dots, x_i^{(n)})$, $i = 1, \dots, n + 1$, які утворюють симплекс n -мірного простору. У цих точках обчислюються значення функції: $f_1 = f(x_1), f_2 = f(x_2), \dots, f_{n+1} = f(x_{n+1})$.

2. **«Сортування»**. З вершин симплекса вибираються три точки: x_h з найбільшим (з обраних) значенням функції f_h , x_g з наступним за величиною значенням f_g та x_l з найменшим значенням функції f_l . Метою подальших маніпуляцій буде зменшення принаймні f_h .

3. Знаходиться **центр ваги** всіх точок, за винятком x_h : $x_c = \frac{1}{n} \sum_{i \neq h} x_i$. Обчислювати $f_c = f(x_c)$ не обов'язково.

4. **«Відображення»**. Відображується точка x_h відносно x_c з коефіцієнтом α (при $\alpha = 1$ це буде центральна симетрія, в загальному випадку – гомотетія), отримаємо точку x_r та обчислимо в ній функцію: $f_r = f(x_r)$. Координати нової точки обчислюються за формулою: $x_r = (1 + \alpha)x_c - \alpha x_h$.

5. Далі залежно від того, наскільки вдалося зменшити функцію, здійснюється пошук місця f_r в ряді f_h, f_g, f_l .

- Якщо $f_r < f_l$, то напрямок вибрано вдало і можна спробувати збільшити крок. Робиться «розтягнення». Нова точка $x_e = (1 - \gamma)x_c + \gamma x_r$ і значення функції $f_e = f(x_e)$.
- Якщо $f_e < f_r$, то можна розширити симплекс до цієї точки: присвоюється точці x_h значення x_e і закінчується ітерація (на крок 9).
- Якщо $f_r < f_e$, то переміщення занадто велике: присвоюється точці x_h значення x_r і закінчується ітерація (на крок 9).
- Якщо $f_l < f_r < f_g$, то вибір точки непоганий (нова краще двох колишніх). Надається точці x_h значення x_r і виконується перехід на крок 9.
- Якщо $f_g < f_r < f_h$, то міняємо місцями значення x_r та x_h . Також потрібно поміняти місцями значення f_r та f_h . Потім виконується перехід на крок 6.

- Якщо $f_h < f_r$, то також виконується перехід на наступний крок 6.

В результаті $f_l < f_g < f_h < f_r$.

6. «Стиснення». Будується точка $x_s = \beta x_h + (1 - \beta)x_c$ і обчислюється в ній значення $f_s = f(x_s)$.

7. Якщо $f_s < f_h$, то привласнюється точці x_h значення x_s і виконується перехід на крок 9.

8. Якщо $f_s > f_h$, то початкові точки виявилися найбільш вдалими. Виконується «глобальне стиснення» симплекса – гомотетія до точки з найменшим значенням $x_l: x_i \leftarrow x_l + \frac{x_i - x_l}{2}, i \neq l$.

9. **Останній крок – перевірка збіжності.** Може виконуватися по-різному, наприклад, за допомогою оцінки дисперсії набору точок. Суть кроку полягає в тому, щоб перевірити взаємну близькість отриманих вершин симплекса, що передбачає і близькість їх до шуканого мінімуму.

Якщо необхідна точність ще не досягнута, виконуються ітерації, починаючи з кроку 2.

На рис. 8.2 зображена блок-схема пошуку мінімуму функції методом деформованого багатогранника.

Рекомендована література та інформаційні ресурси

1. Kolda T. G. Optimization by direct search: new perspectives on some classical and modern methods / T. G. Kolda, R. M. Lewis, V. Torczon // SIAM Review. – 2003. – Vol. 45, No. 3. – P. 385 - 482.
2. Таха Х. Введение в исследование операций. 7-е издание.: Пер. с англ. / Х. Таха. – М.: Издательский дом «Вильямс», 2005. – 912 с.
3. Ржевський С.В. Дослідження операцій / С.В. Ржевський, В.М.Александрова. – К.: Академвидав, 2006. – 560 с.

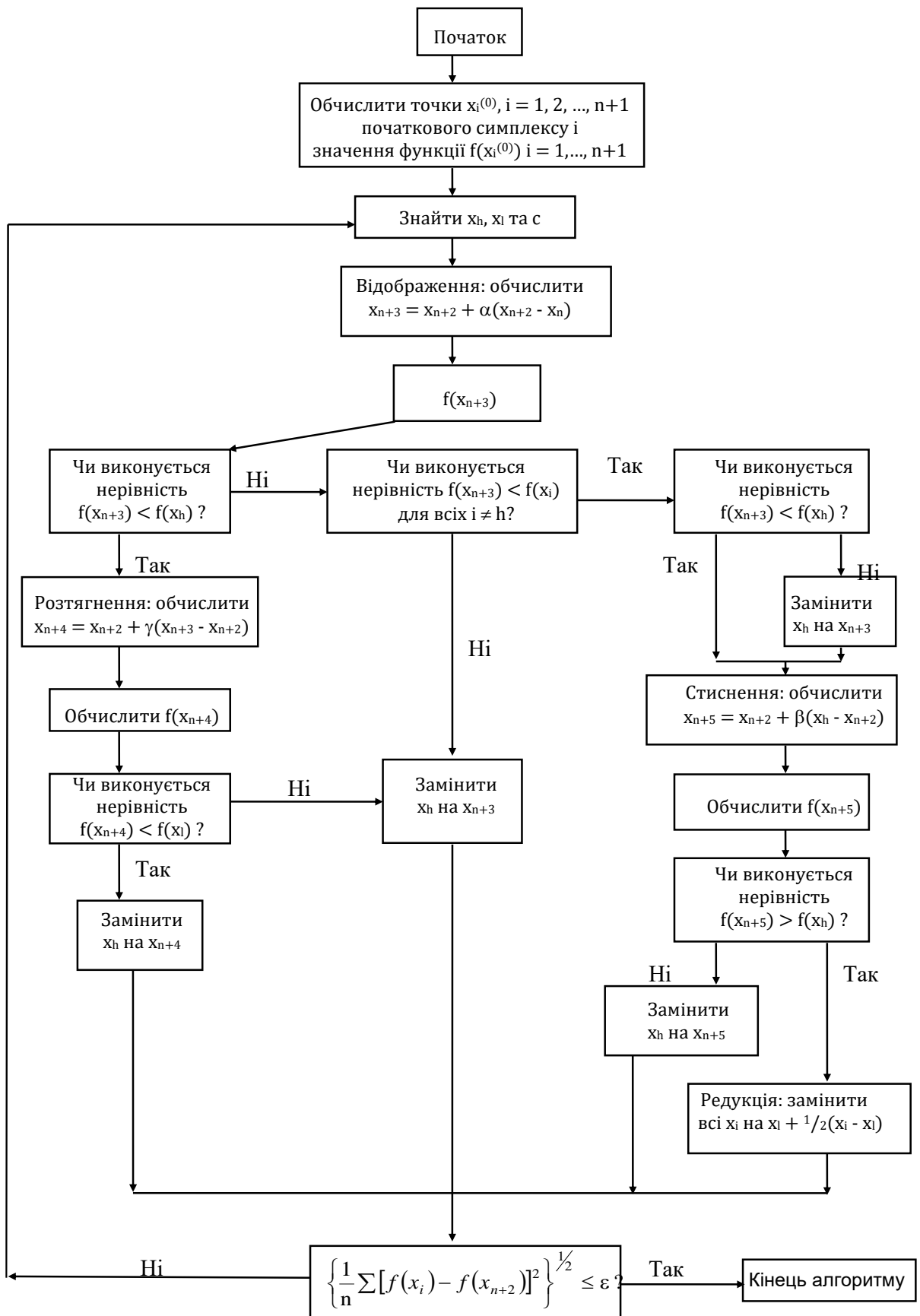


Рис. 8.2. Блок-схема методу пошуку за деформованим багатогранником

Контрольні питання

1. Сформулюйте математичну постановку задачі багатовимірної умовної оптимізації?
2. Назвіть умови оптимальності в задачі безумовної багатовимірної оптимізації?
3. Які обчислювальні методи називають пасивними; послідовними?
4. Дайте визначення мінімізаційної та релаксаційної послідовності точок відносно заданої функції?
5. Наведіть загальну ітераційну формулу методів спуску.
6. Що визначає порядок релаксаційного методу?
7. Яку інформацію використовують для вибору напрямку спуску в методах першого порядку?
8. Сформулюйте ідею методу покоординатного спуску.
9. Яким чином здійснюється операція стиснення в методі пошуку за деформованим багатогранником?
10. Які критерії зупинки ітераційного процесу застосовують в задачах умовної багатовимірної оптимізації?
11. Які критерії зупинки ітераційного процесу використовують в задачах безумовної багатовимірної оптимізації?
12. Наведіть визначення лінійної швидкості збіжності методу спуску.
13. Яке співвідношення визначає надлінійну, квадратичну швидкість збіжності ітераційного процесу методу спуску?
14. Назвіть переваги і недоліки прямих методів багатовимірної безумовної оптимізації?
15. Що є критерієм зупинки ітераційного процесу в методі пошуку за деформованим багатогранником?

Завдання для самостійної та індивідуальної роботи

Знайти мінімум функції відповідно до варіанта, використовуючи метод покоординатного спуску і метод деформованого багатогранника. Методи реалізувати у вигляді комп'ютерної програми у будь-якому середовищі на вибір студента. Надати геометричну інтерпретацію процесу розв'язання задачі.

Варіанти завдань

1. $f(x) = x_1^2 + x_2^2 + x_1 - x_1x_2 - 2x_2$
2. $f(x) = (x_2 - x_1^2)^2 + 100(1 - x_1)^2$
3. $f(x) = 5(x_2 - x_1^2)^2 + (4 - x_1x_2)^2$
4. $f(x) = x_1^2 + 25x_2^2 + 3x_1x_2 - 106x_1 - 32x_2$
5. $f(x) = 5x_1^2 + 14x_2^2 + 9x_1x_2 - 27x_1 - 82x_2 + 33$
6. $f(x) = 4x_1^2 + x_2^2 - x_1x_2 + 8x_1 - 12x_2 + 10$
7. $f(x) = 81(x_2 + x_1^2)^2 + (12 - x_1)^2$
8. $f(x) = (3x_2 - 2x_1^2)^2 + 4(2 - x_1x_2)^2$
9. $f(x) = 5x_1^2 + 9x_2^2 + 2x_1x_2 - 16x_1 - 7x_2 + 90$
10. $f(x) = 16x_1^2 + 25x_2^2 + 13x_1x_2 - 42x_1 - 18x_2 + 25$
11. $f(x) = x_1^2 + 4x_2^2 + x_3^2 + x_1 - 2x_2 - 7x_3 + 43$
12. $f(x) = 4(x_2 - x_1^2)^2 + 64(1 - x_2)^2$
13. $f(x) = (x_2 - x_1^2)^2 + 49(1 - x_1x_2)^2$
14. $f(x) = 10x_1^2 + x_2^2 + 4x_1x_2 - 28x_1 + 32x_2 + 66$
15. $f(x) = x_1^2 + 14x_2^2 - 5x_1x_2 + 17x_1 - 2x_2 + 3$
16. $f(x) = 3x_1^2 + 24x_2^2 + x_3^2 + 8x_1x_2 - 7x_2 - 12x_3 + 35$
17. $f(x) = 81(x_2 - x_1^2)^2 + (6 - x_2)^2$
18. $f(x) = (x_1 - x_2^2)^2 + (9 - x_1x_2)^2$
19. $f(x) = 25x_1^2 + 64x_2^2 + 14x_1x_2 - 160x_1 - 45x_2 + 100$
20. $f(x) = 9x_1^2 + 16x_2^2 - 2x_1x_2 - 17x_1 - 18x_2 + 19$
21. $f(x) = x_1^2 + 49x_2^2 + 5x_1x_2 - 8x_1 + 90x_2 + 83$
22. $f(x) = 100(x_2 - x_1^2)^2 + (4 + x_2)^2$
23. $f(x) = (x_2 - x_1^2)^2 + 36(2 - x_1x_2)^2$
24. $f(x) = 9x_1^2 + 25x_2^2 + 14x_1x_2 - 35x_1 - 82x_2 + 773$
25. $f(x) = 5x_1^2 + 14x_2^2 - 9x_1x_2 - 27x_1 - 55x_2 + 59$
26. $f(x) = 4x_1^2 + 49x_2^2 + 11x_1x_2 + 12x_1 - 44x_2 + 69$
27. $f(x) = 16(x_1 + x_2^2)^2 + (4x_1 + x_2)^2$
28. $f(x) = 49(x_2 - x_1^2)^2 + (25 + x_1x_2)^2$
29. $f(x) = 9x_1^2 + 25x_2^2 + 12x_1x_2 - 72x_1 + 35x_2 + 235$
30. $f(x) = 15x_1^2 + 24x_2^2 + 13x_1x_2 - 47x_1 - 8x_2 + 5$

Практичне заняття № 9

Методи нульового порядку: Розенброка, Хука і Дживса

Мета: засвоїти основні ідеї методів нульового порядку: Хука і Дживса, Розенброка для розв'язання задач безумовної оптимізації

План заняття

1. Постановка задачі багатовимірної безумовної оптимізації
2. Метод конфігурацій (Алгоритм Хука і Дживса)
3. Метод Розенброка

Опорні поняття

1. Постановка задачі багатовимірної безумовної оптимізації

Будемо розглядати задачу багатовимірної безумовної оптимізації у наступній постановці:

$$f(x) \rightarrow \min_{x \in R^n}. \quad (9.1)$$

2. Метод конфігурацій (Алгоритм Хука і Дживса)

У цьому алгоритмі пропонується логічно проста стратегія пошуку, яка використовує апріорні відомості про топологію функції і, водночас, відкидає вже застарілу інформацію про цю функцію. В інтерпретації Вуда алгоритм включає два основні етапи:

- 1) досліджувальний пошук навколо базисної точки \bar{x}^k ;
- 2) пошук за "зразком", тобто у напрямку, обраному для мінімізації.

Алгоритм починається із задання початкової точки пошуку \bar{x}^0 та початкового приросту (кроку) $\Delta\bar{x}^0$. Після цього проводиться досліджувальний пошук,

Досліджувальний пошук. Робимо пробний крок за змінною тобто визначаємо точку $\bar{x}_1^0 + \Delta\bar{x}_1^0$ і обчислюємо значення функції в точці $\bar{x}' = (\bar{x}_1^0 + \Delta\bar{x}_1^0, \dots, \bar{x}_n^0)$.

Якщо значення функції в даній точці більше за $f(\bar{x}^0)$, то здійснюємо пробний крок за цією ж самою змінною, але у протилежному напрямку. Якщо значення функції і в точці $\bar{x}'' = (\bar{x}_1^0 - \Delta\bar{x}_1^0, \dots, \bar{x}_n^0)$ є більшим за $f(\bar{x}^0)$, то залишаємо точку \bar{x}_1^0 без змін. Інакше замінюємо точку \bar{x}^0 на \bar{x}' або на \bar{x}'' залежно від того, де значення функції менше за вихідне. З отриманої точки робимо пробні кроки за рештою змінних, використовуючи той самий алгоритм.

Якщо у процесі досліджувального пошуку не вдається зробити жодного вдалого пробного кроку, то $\Delta\bar{x}$ необхідно скорегувати (зменшити). Після чого знову переходимо до досліджувального пошуку.

Якщо у процесі досліджувального пошуку зроблений хоча б один вдалий пробний крок, то переходимо до пошуку за зразком.

Пошук за зразком. Після досліджувального пошуку отримуємо точку \bar{x}^{01} . Вектор $\bar{x}^{01} - \bar{x}^0$ визначає напрямок, в якому функція зменшується. Тому проводимо мінімізацію функції за цим напрямком, розв'язуючи таку задачу:

$$\min_{\lambda} f(\bar{x}^0 + \lambda(\bar{x}^{01} - \bar{x}^0)).$$

При проведенні пошуку за зразком величина кроку за кожною змінною пропорційна величині кроку на етапі досліджувального пошуку. Якщо вдається зробити вдалий крок у пошуку за зразком, то як результат знаходимо нове наближення $\bar{x}^1 = \bar{x}^{01} + \lambda^0(\bar{x}^{01} - \bar{x}^0)$, де $\lambda^0 = \arg \min_{\lambda} f(\bar{x}^{01} + \lambda(\bar{x}^{01} - \bar{x}^0))$.

З точки \bar{x}^1 починаємо новий досліджувальний пошук і т.д. (рис. 9.1)

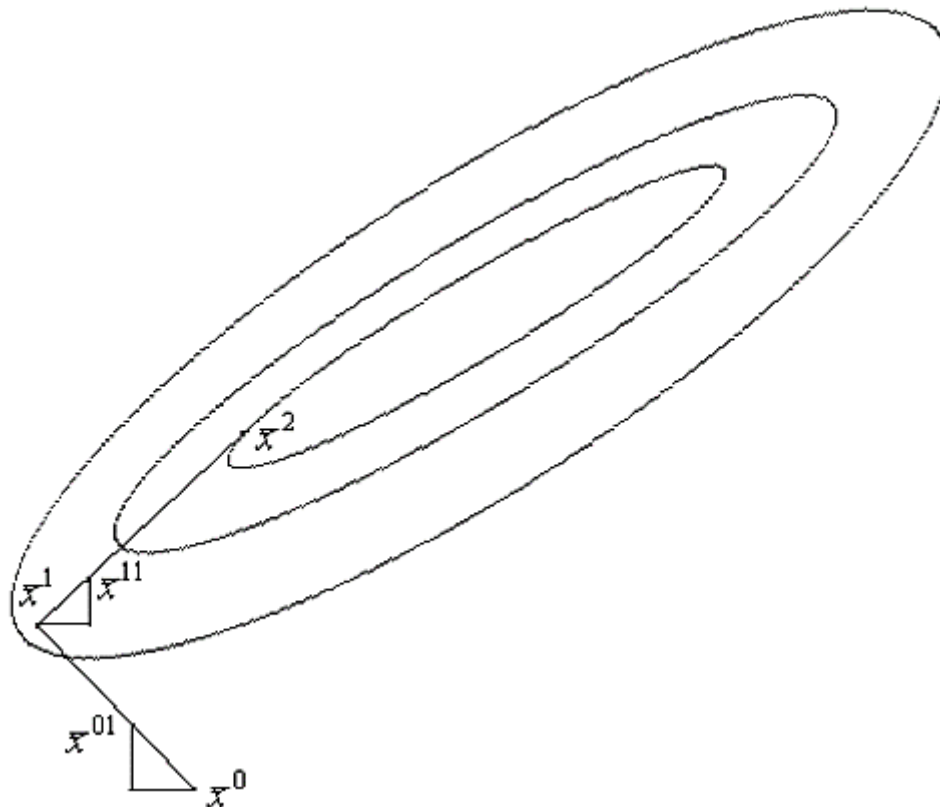


Рис. 9.1. Приклад траєкторії спуску в алгоритмі Хука і Дживса

Можливі модифікації алгоритму, в яких у процесі досліджувального пошуку визначають мінімум за кожною змінною або у процесі пошуку за зразком знаходять не мінімум функції, а просто роблять крок у знайденому напрямку з фіксованим значенням параметра λ .

3. Метод Розенброка

Цей ітераційний метод має деяку схожість з алгоритмом Хука та Дживса. Його називають також *методом обертових координат*. Він помітно ефективніший за попередні, особливо при мінімізації функцій яружного типу.

Загальна ідея методу у тому, що вибирається система ортогональних напрямків $\bar{S}_1^0, \bar{S}_2^0, \dots, \bar{S}_n^0$, у кожному з яких послідовно шукається мінімальне значення, після чого система напрямків повертається так, щоб одна з осей збіглася із напрямком повного переміщення, інші були ортогональні між собою.

Припустимо \bar{x}^0 – вектор початкового наближення; $\bar{S}_1^0, \bar{S}_2^0, \dots, \bar{S}_n^0$ – система ортогональних напрямків. На першій ітерації це може бути ортонормована система координат. Починаючи з \bar{x}^0 , послідовно здійснюємо мінімізацію функції $f(\bar{x})$ у напрямках $\bar{S}_1^0, \bar{S}_2^0, \dots, \bar{S}_n^0$, знаходячи послідовні наближення, а саме:

$$\begin{aligned}\bar{x}_1^0 &= \bar{x}_0^0 + \lambda_1 \bar{S}_1^0, & \lambda_1 &= \arg \min_{\lambda} f(\bar{x}_0^0 + \lambda \bar{S}_1^0), \\ \bar{x}_2^0 &= \bar{x}_1^0 + \lambda_2 \bar{S}_2^0, & \lambda_2 &= \arg \min_{\lambda} f(\bar{x}_1^0 + \lambda \bar{S}_2^0), \\ & & & \dots \\ \bar{x}_n^0 &= \bar{x}_{n-1}^0 + \lambda_n \bar{S}_n^0, & \lambda_n &= \arg \min_{\lambda} f(\bar{x}_{n-1}^0 + \lambda \bar{S}_n^0).\end{aligned}$$

Наступна ітерація починається з точки $\bar{x}^1 = \bar{x}_n^0$.

Якщо не змінити систему напрямків, то алгоритм буде повторювати алгоритм Гаусса. Тому після завершення чергового k -го етапу обчислюємо нові напрямки пошуку. Ортогональні напрями пошуку повертають так, аби вони виявилися витягнутими вздовж "яру" ("хребта") і, таким чином, виключатиметься взаємодія змінних $(x_i x_j)$. Отже, напрямки пошуку витягуються вздовж основних осей квадратичної апроксимації цільової функції (рис. 9.2).

Розглянемо k -е ітерацію алгоритму Розенброка. В результаті мінімізації за кожним з ортогональних напрямків ми маємо на даній ітерації систему параметрів $\lambda_1^k, \lambda_2^k, \dots, \lambda_n^k$, за допомогою якої визначимо систему векторів $\bar{A}_1^k, \bar{A}_2^k, \dots, \bar{A}_n^k$, обчислюваних за формулами наступного вигляду

$$\begin{aligned}\bar{A}_1^k &= \lambda_1^k \bar{S}_1^k + \lambda_2^k \bar{S}_2^k + \dots + \lambda_n^k \bar{S}_n^k; \\ \bar{A}_2^k &= \lambda_2^k \bar{S}_2^k + \dots + \lambda_n^k \bar{S}_n^k; \\ & \dots \\ \bar{A}_n^k &= \lambda_n^k \bar{S}_n^k.\end{aligned}$$

Ця система дозволяє побудувати нову систему ортогональних напрямів $\bar{S}_1^{k+1}, \bar{S}_2^{k+1}, \dots, \bar{S}_n^{k+1}$. Причому перший вектор спрямовується так, аби він збігався з напрямком загального переміщення на k -му кроці, а решта напрямків отримується за допомогою процедури ортогоналізації Грама – Шмідта, тобто

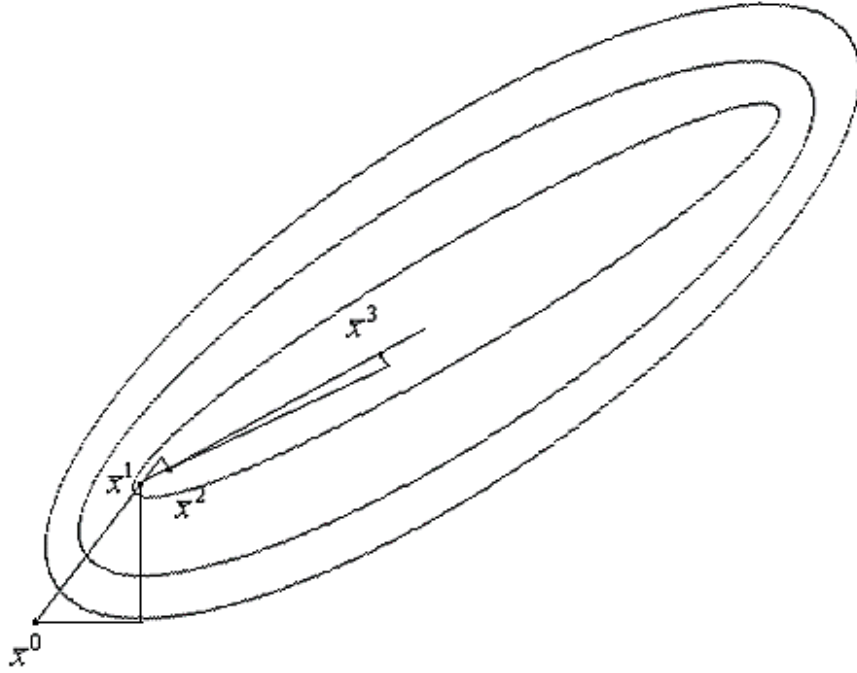


Рис. 9.2. Приклад траєкторії спуску в алгоритмі Розенброка

$$\begin{aligned}\bar{S}_1^{k+1} &= \frac{\bar{A}_1^k}{\|\bar{A}_1^k\|}; \\ \bar{B}_2^k &= \bar{A}_2^k - [(\bar{A}_2^k)^T \bar{S}_1^{k+1}] \bar{S}_1^{k+1}; \\ \bar{S}_2^{k+1} &= \frac{\bar{B}_2^k}{\|\bar{B}_2^k\|}; \\ \bar{B}_l^k &= \bar{A}_l^k - \sum_{m=1}^{l-1} [(\bar{A}_l^k)^T \bar{S}_m^{k+1}] \bar{S}_m^{k+1}; \\ \bar{S}_l^{k+1} &= \frac{\bar{B}_l^k}{\|\bar{B}_l^k\|}; l = 2, 3, \dots, n.\end{aligned}$$

Для ефективної роботи алгоритму необхідно, аби жоден з векторів системи $\bar{S}_1^{k+1}, \bar{S}_2^{k+1}, \dots, \bar{S}_n^{k+1}$ не виявився нульовим вектором. Для цього в алгоритмі слід вишукувати параметри $\lambda_1^k, \lambda_2^k, \dots, \lambda_n^k$ за спаданням абсолютного значення, тобто $|\lambda_1^k| > |\lambda_2^k| > \dots > |\lambda_n^k|$. Тоді якщо будь-які m з λ_i^k перетворюються на нуль, то відшукуються нові напрямки \bar{S}_l^{k+1} тільки для тих $(n - m)$ напрямків, для яких $\lambda_i^k \neq 0$. Решта m напрямків залишаються незмінними: $\bar{S}_l^{k+1} = \bar{S}_l^k, l = \overline{(n - m + 1), n}$. Оскільки перші $(n - m)$ векторів є взаємно ортогональними, $\lambda_i^k = 0, i = \overline{(n - m + 1), n}$, і перші $(n - m)$ векторів не матимуть складових у напрямках $\bar{S}_l^{k+1}, l = \overline{(n - m + 1), n}$. А з взаємної ортогональності цих останніх напрямків випливає, що всі напрямки є взаємно ортогональними.

Палмером було показано, що \bar{B}_{l+1}^k і $\|\bar{B}_{l+1}^k\|$ є пропорційними λ_l^k (за умови $\sum_{i=l}^n (\lambda_i^k)^2 \neq 0$). Отже, у процесі обчислення $\bar{S}_l^{k+1} = \frac{\bar{B}_l^k}{\|\bar{B}_l^k\|}$ величина λ_l^k зменшується, і таким чином, напрямок \bar{S}_l^{k+1} залишається визначеним, навіть якщо $\lambda_l^k = 0$. Враховуючи це, Палмер запропонував для обчислення \bar{S}_l^{k+1} наступні формули:

$$\bar{A}_i^k = \sum_{m=i}^n \lambda_m^k \bar{S}_m^k, i = 1, 2, \dots, n;$$

$$\bar{S}_i^{k+1} = \frac{\bar{A}_i^k \|\bar{A}_{i-1}^k\|^2 - \bar{A}_{i-1}^k \|\bar{A}_i^k\|^2}{\|\bar{A}_{i-1}^k\| \cdot \|\bar{A}_i^k\| \cdot [\|\bar{A}_{i-1}^k\|^2 - \|\bar{A}_i^k\|^2]^{1/2}}; i = 2, 3, \dots, n.$$

$$\bar{S}_1^{k+1} = \frac{\bar{A}_1^k}{\|\bar{A}_1^k\|}.$$

Критерії зупинки алгоритму можуть бути стандартними (вони описані у попередніх алгоритмах прямих методів).

Варіанти завдань для самостійної та індивідуальної роботи

Реалізувати алгоритм Хука і Дживса або метод Розенброка у вигляді програми у будь-якому середовищі, знайти мінімум функції, наведеної у варіантах завдань для практичного заняття № 8. Надати геометричну інтерпретацію процесу розв'язання задачі.

Рекомендована література та інформаційні ресурси

1. Kolda T. G. Optimization by direct search: new perspectives on some classical and modern methods / T. G. Kolda, R. M. Lewis, V. Torczon // SIAM Review. – 2003. – Vol. 45, No. 3. – P. 385–482.
2. Таха Х. Введение в исследование операций. 7-е издание.: Пер. с англ. / Х. Таха. – М.: Издательский дом «Вильямс», 2005. – 912 с.
3. Ржевський С.В. Дослідження операцій / С.В. Ржевський, В.М.Александрова. – К.: Академвидав, 2006. – 560 с.

Контрольні питання

1. Опишіть ідею методу Гаусса, переваги, недоліки, критерії зупинки.
2. Опишіть етапи алгоритма Хука та Дживса.
3. Яким чином в алгоритмі Розенброка здійснюється поворот системи координат? Для чого в цьому алгоритмі використовують ортогоналізацію Грама – Шмідта?
4. Опишіть симплексний метод Нелдера – Міда, його ідею, етапи алгоритму пошуку за деформованим багатогранником?
5. Які критерії зупинки ітераційних процесів застосовуються в методах нульового порядку?

Практичне заняття № 10

Гرادієнтні методи безумовної оптимізації

Мета: засвоїти ідею та умови застосування градієнтного методу, різних його модифікацій. Умови збіжності методу

План заняття

1. Постановка задачі безумовної оптимізації
2. Основна властивість градієнта. Градієнтні методи розв'язання задач багатовимірної безумовної мінімізації
3. Умови збіжності методів
4. Спеціальні класи негладких функцій. Джерела, що породжують задачі негладкої оптимізації

Опорні поняття

1. Постановка задачі безумовної оптимізації

Будемо розглядати задачу

$$f(x) \rightarrow \max_{x \in R^n}, \quad f(x) \in C^1(R^n). \quad (10.1)$$

або

$$f(x) \rightarrow \min_{x \in R^n}, \quad f(x) \in C^1(R^n). \quad (10.2)$$

2. Основна властивість градієнта. Градієнтні методи розв'язання задач багатовимірної безумовної мінімізації

Чільне місце серед прямих методів розв'язання екстремальних задач займає *градієнтний метод* (точніше, сімейство градієнтних методів) пошуку стаціонарних точок диференційованої функції $f(x)$.

Нагадаємо, *стаціонарною* називається точка, в якій $\nabla f(x) = 0$, і що згідно з необхідними умовами оптимальності є «підозрілою» стосовно наявності локального екстремуму. Відтак, застосовуючи градієнтний метод, знаходять множину точок локальних мінімумів (або максимумів), серед яких визначається мінімум (або максимум) глобальний.

Ідея даного методу заснована на властивості градієнта функції вказувати напрямок її найбільш швидкого зростання в околі точки, в якій він обчислений. Тому, якщо з певної поточної точки $x^{(1)}$ переміщуватися в напрямку вектора $\nabla f(x) = 0$, то функція f буде зростати, принаймні, в деякому околі точки $x^{(1)}$.

Відтак, для точки $x^{(2)} = x^{(1)} + \lambda \nabla f(x^{(1)})$, ($\lambda > 0$), що лежить в такому околі, справедливою є нерівність $f(x^{(1)}) \leq f(x^{(2)})$ (рис. 10.1).

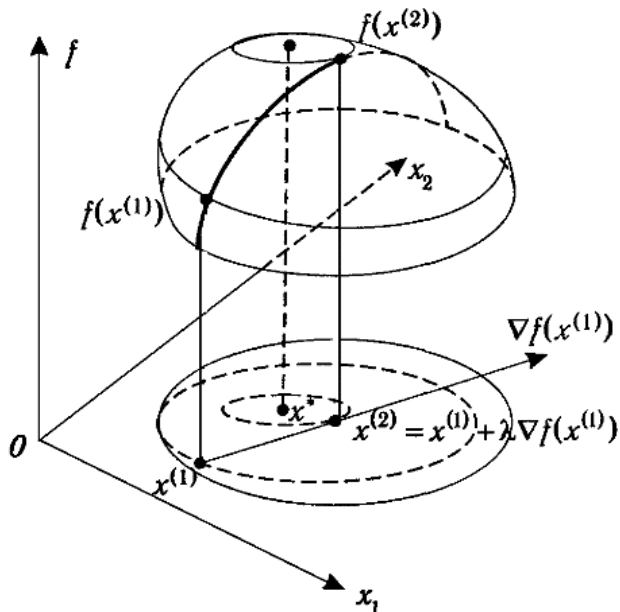


Рис. 10.1. Напрямок підйому в точці $x^{(1)}$

Розглянемо задачу (10.2). Як тільки визначається напрямок руху, відразу ж постає питання про те, як далеко слід рухатися в цьому напрямку. Іншими словами, виникає проблема вибору кроку λ у рекурентній формулі

$$x^{(q+1)} = x^{(q)} - \lambda \nabla f(x^{(q)}), \quad (10.3)$$

що задає послідовність точок, які прагнуть до точки мінімуму цільової функції. Деякі способи одновимірного пошуку параметра довжини кроку описані в Практичному занятті № 8.

Вибір крокового множника визначає різні модифікації градієнтного методу. Найбільш відомим є *метод найшвидшого спуску*.

Метод найшвидшого спуску. Нехай $f(x) = f(x_1, x_2, \dots, x_n)$ – диференційовна функція, задана на R^n , а $x^{(q)} = (x_1^{(q)}, x_2^{(q)}, \dots, x_n^{(q)})$ – деяка поточна точка. До речі, будь-яких загальних рекомендацій щодо вибору вихідної точки (початкового наближення) $x^{(0)}$ не існує. По можливості, вона мусить знаходитися близько від шуканого оптимального плану x^* . Якщо $x^{(q)}$ не є стаціонарною точкою (тобто $|\nabla f(x^{(q)})| > 0$), то, рухаючись у напрямку *антиградієнта* $-\nabla f(x^{(q)})$, функція $f(x)$ на деякому проміжку обов'язково буде спадати. Звідси виникає природна ідея такого вибору кроку, щоб рух у зазначеному напрямку продовжувався, доки зростання не припиниться. Для цього висловимо залежність значення $f(x)$ від крокового множника $\lambda > 0$, вважаючи $x = x^{(q)} - \lambda \nabla f(x^{(q)})$:

$$f(x) = f\left(x^{(q)} - \lambda \nabla f(x^{(q)})\right) = \phi(\lambda), \quad (10.4)$$

або в координатній формі

$$\phi(\lambda) = f\left(x_1^{(q)} - \lambda \frac{\partial f(x^{(q)})}{\partial x_1}, \dots, x_n^{(q)} - \lambda \frac{\partial f(x^{(q)})}{\partial x_n}\right). \quad (10.5)$$

Аби досягти найбільшого з можливих значень $f(x)$ під час руху у напрямку $-\nabla f(x^{(q)})$, потрібно вибрати таке значення $\tilde{\lambda}$, яке мінімізує функцію $\phi(\lambda)$ ($\phi(\tilde{\lambda}) = \min_{\lambda > 0} \phi(\lambda)$). Для обчислення $\tilde{\lambda}$ можна скористатися необхідною умовою екстремуму: $d\phi(\tilde{\lambda})/d\lambda = 0$. Помітимо, що коли для будь-якого $\lambda > 0$ виконується нерівність $d\phi(\tilde{\lambda})/d\lambda < 0$, функція $f(x)$ не є обмеженою знизу (тобто не має мінімуму). У протилежному випадку на основі (5) отримаємо

$$\frac{d\phi}{d\lambda} = \frac{\partial f(x)}{\partial x_1} \times \frac{dx_1}{d\lambda} + \dots + \frac{\partial f(x)}{\partial x_n} \times \frac{dx_n}{d\lambda}, \quad (10.6)$$

що, в свою чергу, дає

$$\frac{d\phi}{d\lambda} = \frac{\partial f(x)}{\partial x_1} \times \frac{\partial f(x^{(q)})}{\partial x_1} + \dots + \frac{\partial f(x)}{\partial x_n} \times \frac{\partial f(x^{(q)})}{\partial x_n} = \nabla f(x) \nabla f(x^{(q)}). \quad (10.7)$$

Якщо вважати, що наступна точка $x^{(q+1)}$ відповідає оптимальному значенню $\lambda = \tilde{\lambda}$, то в ній має виконуватися рівність $d\phi(\tilde{\lambda})/d\lambda = 0$, і $\tilde{\lambda}$ варто знаходити з умови

$$\nabla f\left(x^{(q)} - \tilde{\lambda} \nabla f(x^{(q)})\right) \nabla f(x^{(q)}) = 0. \quad (10.8)$$

Умова (10.8) означає рівність нулю скалярного добутку градієнтів функції f в точках $x^{(q+1)}$ і $x^{(q)}$. Геометрично вона може бути інтерпретована як перпендикулярність векторів градієнтів функції f у вказаних точках, що й наведено на рис. 10.2. В точці $x^{(q+1)}$ вектор $-\nabla f(x^{(q+1)})$, будучи антиградієнтом, є перпендикулярним лінії рівня, яка проходить через дану точку, а вектор $-\nabla f(x^{(q)})$ є дотичним до цієї лінії.

Рух у напрямку антиградієнта $-\nabla f(x^{(q)})$ слід продовжувати, допоки він перетинає лінії рівня оптимізованої функції. Після того, як точку $x^{(q+1)}$ знайдено, вона стає поточною для чергової ітерації. Насправді, ознакою досягнення стаціонарної точки слугує досить мала зміна координат точок, котрі розглядаються на послідовних ітераціях. Одночасно з цим координати вектора $\nabla f(x^{(q)})$ мають бути близькими до нуля.

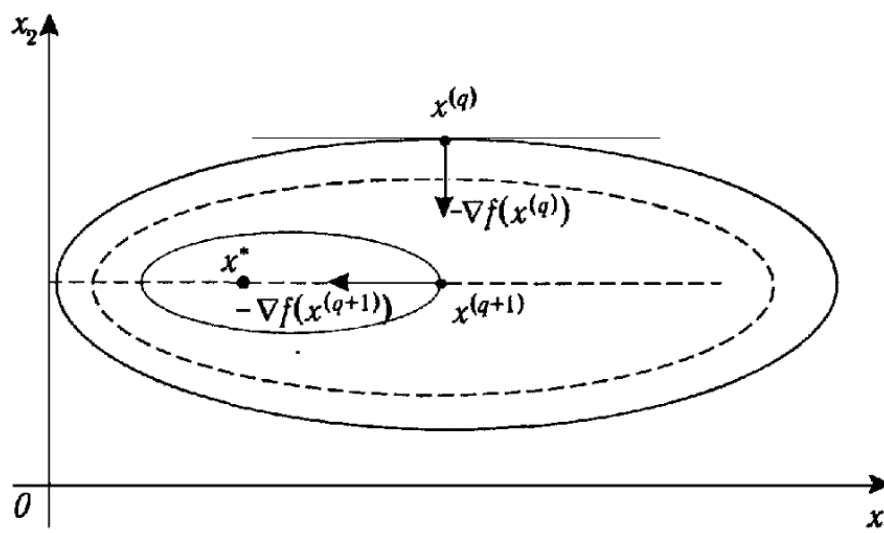


Рис. 10.2. Геометрична інтерпретація методу найшвидшого спуску

Алгоритм методу найшвидшого спуску

1. Задаємо точність обчислень $\varepsilon > 0$, величину $\bar{\lambda} > 0$ і початкове наближення $x^{(0)}, k = 0$.
2. Обчислюємо значення цільової функції $f(x^{(0)})$.
3. Запам'ятовуємо поточну точку $x^{(k)}$, обчислюємо градієнт функції в цій точці і його норму.
4. Перевіряємо умову $\|\nabla f(x^{(k)})\| < \varepsilon$. Якщо вона виконується, то переходимо на крок 8, інакше – на крок 5.
5. Здійснюємо процедуру одновимірного пошуку параметра довжини кроку λ_k , мінімізуючи функцію (10.5) за умови $\lambda \in [0, \bar{\lambda}]$. Для цього можна скористатися методом одновимірної оптимізації (див. Практичне заняття № 5).
6. Обчислюємо нове наближення за формулою

$$x^{(k+1)} = x^{(k)} - \lambda_k \nabla f(x^{(k)}).$$

7. $k := k + 1$. Перехід на крок 3.

8. $x^* = x^k, f(x^*) = f(x^k)$.

Кінець алгоритму.

Зауваження. Ітераційний процес можна також завершити, коли відстань між двома послідовними наближеннями виявиться меншою за точність, тобто:

$$\sqrt{\sum_{i=1}^n (x_i^{k+1} - x_i^k)^2} < \varepsilon.$$

Градієнтний метод з дробленням кроку. Для знаходження кроку λ в методі найшвидшого спуску потрібно розв'язати рівняння (10.8), яке може бути досить складним. Тому часто обмежуються «підбором» такого значення λ , аби

$\phi(\lambda) < \phi(0)$. Для цього задаються деяким початковим значенням λ_1 (наприклад, $\lambda_1 = 1$) і перевіряють умову

$$\phi(\lambda_1) < \phi(0). \quad (10.9)$$

Якщо вона не виконується, то вважають, що $\lambda_2 = \beta \cdot \lambda_1$, де $0 < \beta < 1$ – параметр методу, який називається коефіцієнтом дроблення, і т. д. допоки не вдається знайти відповідний крок, з яким переходять до наступної точки $x^{(q+1)}$.

Критерій завершення алгоритму, очевидно, буде таким самим, як і в методі найшвидшого спуску.

Алгоритм градієнтного методу з дробленням кроку

1. Задаємо параметр точності $\varepsilon > 0$, початковий крок $\bar{\alpha}$, початкове наближення $X^0 \in E^n$, обчислюємо значення $f(X^0)$, вважаємо, що $k = 0$.

2. Обчислюємо градієнт $f'(X^k)$ і його норму $\|f'(X^k)\|$, перевіряємо умову $\|f'(X^k)\| < \varepsilon$. Якщо умова виконується, переходимо до п. 5, інакше – до п. 3.

3. $\alpha_k = \alpha$. Знаходимо нову точку $X^{(k+1)}$ за формулою

$$X^{(k+1)} = X^k - \alpha_k f'(X^k)$$

і обчислюємо $f(X^{k+1})$.

4. Якщо $f(X^{k+1}) < f(X^k)$, то вважаємо, що $X^k = X^{(k+1)}$, $f(X^k) = f(X^{k+1})$, $\alpha_k = 1,25\alpha_k$, $k = k + 1$ й переходимо до п. 2, інакше $\alpha_k = 0,5\alpha_k$ і переходимо до п. 3.

5. $X^* = X^k$, $f^* = f(X^k)$.

Кінець алгоритму.

Градієнтний метод з адаптивним вибором кроку. Реалізується аналогічно попередньому методу, але замість умови (10.9) перевіряється подана нижче, а саме:

$$\phi(\lambda_1) - \phi(0) = f(x^{(q)} - \lambda_1 f'(x^{(q)})) - f(x^{(q)}) \leq -\varepsilon_0 \lambda_1 \cdot \|\nabla f(x^{(q)})\|^2.$$

Градієнтний метод з апіорним вибором кроку відрізняється від описаних вище тим, що на кожній q – й ітерації крок λ задається з таких умов:

$$\sum_{q=0}^{\infty} \lambda_q = \infty, \quad \sum_{q=0}^{\infty} (\lambda_q)^2 < \infty.$$

Наприклад,

$$\lambda_q = \frac{1}{q+1}.$$

3. Умови збіжності градієнтних методів

Наведемо теореми про збіжність методів градієнтного спуску.

Теорема 10.1. Нехай функція $f(x)$ є обмеженою знизу на E^n ; тобто $f_* = \inf_{x \in E^n} f(x) > -\infty$; $f(x) \in C^1(E^n)$; і її градієнт задовольняє умову Ліпшиця з константою Ліпшиця $L > 0$, а саме:

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|, \quad x, y \in E^n.$$

Тоді послідовність точок $\{x^{(q)}\}$, побудована методом найскорішого спуску, починаючи з довільного початкового наближення $x^{(0)}$, є такою, при якій

$$\lim_{q \rightarrow \infty} \|\nabla f(x^{(q)})\| = 0.$$

Якщо при цьому множина $M_0(x^{(0)}) = \{x \in E^n: f(x) \leq f(x^{(0)})\}$ обмежена, то

$$\lim_{q \rightarrow \infty} \rho(x^{(q)}, S_*) = 0,$$

де $S_* = \{x \in M_0(x^{(0)}): \nabla f(x) = 0\}$ – множина стаціонарних точок функції $f(x)$ на множині $M_0(x^{(0)})$.

Теорема 10.2. Нехай функція $f(x)$ є неперервно диференційовною на E^n : $f(x) \in C^1(E^n)$. Тоді будь-яка гранична точка будь-якої траєкторії $\{x^{(q)}\}$, побудованої за допомогою градієнтного методу з використанням правила Арміхо або правила одновимірної мінімізації, є стаціонарною точкою задачі (10.2). Якщо послідовність $\{x^{(q)}\}$ є обмеженою, то має місце граничне співвідношення

$$\{\nabla f(x^{(q)})\} \rightarrow 0, \quad q \rightarrow \infty.$$

Теорема 10.3. Нехай для задачі (10.2) виконані умови теореми 10.1 і, крім того, функція $f(x)$ є опуклою на E^n : $f(x) \in C^1(E^n)$; Тоді для послідовності $\{x^{(q)}\}$, побудованої за допомогою градієнтного методу з використанням правила Арміхо або правила одновимірної мінімізації, мають місце такі граничні співвідношення:

$$\lim_{q \rightarrow \infty} f(x^{(q)}) = f_*, \quad \lim_{q \rightarrow \infty} \rho(x^{(q)}, X_*) = 0,$$

де $X_* = \left\{x \in E^n: f(x) = f_* = \inf_{x \in E^n} f(x)\right\}$.

Теорема 10.4. Нехай функція $f(x)$ є сильно опуклою на E^n , $f(x) \in C^2(E^n)$; а її матриця других часткових похідних задовольняє умову

$$d\|h\|^2 \leq (f''(x)h, h) \leq D\|h\|^2, \quad x, h \in E^n, \quad 0 < d \leq D < \infty.$$

Тоді побудована за допомогою методу найшвидшого спуску з будь-якого початкового наближення $x^{(0)}$ послідовність точок $\{x^{(q)}\}$ збігається до точки мінімуму x_* функції $f(x)$ зі швидкістю геометричної прогресії, тобто

$$f(x^{(q)}) - f(x_*) \leq m^q (f(x^{(0)}) - f(x_*)),$$

$$\|x^{(q)} - x_*\| \leq C(\sqrt{m})^q,$$

де $m = \frac{D-d}{D+d}$, $C > 0$ – константи.

4. Спеціальні класи негладких функцій. Джерела, що породжують задачі негладкої оптимізації

Розглянемо деякі спеціальні класи недиференційованих функцій, які, з одного боку, є не надто широкими узагальненнями класу неперервно диференційованих функцій, оскільки з них можна досить розумно визначити узагальнене поняття градієнта і використовувати схеми процесів градієнтного типу знаходження екстремуму. З іншого боку, введені спеціальні класи функцій охоплюють і функції, що зустрічаються в більшості практичних додатків, де йдеться про неперервні функції, визначені на n -вимірному евклідовому просторі.

4.1. Поняття субградієнта опуклої функції. Деякі способи обчислення субградієнта

Нехай $f(x)$ – опукла функція, визначена на множині $M \subseteq E_n$, X^* – множина мінімумів (яка може бути і порожньою), $x^* \in X^*$ – точка мінімуму; $\inf f(x) = f^*$; \bar{x} – внутрішня точка області визначення M .

Визначення 10.1 *Субградієнтом (узагальненим градієнтом) $g_f(\bar{x})$ функції f в точці $\bar{x} \in \text{int } M$ називається вектор $g_f(\bar{x})$, при якому*

$$f(x) - f(\bar{x}) \geq (g_f(\bar{x}), x - \bar{x}) \text{ для всіх } x \in M.$$

З визначення субградієнта випливає, що у випадку виконання умови $f(x) < f(\bar{x})$ має місце така нерівність:

$$(-g_f(\bar{x}), x - \bar{x}) > 0. \tag{10.10}$$

Множину узагальнених градієнтів (субградієнтну множину) опуклої функції $f(x)$ в точці \bar{x} будемо позначати $G_f(\bar{x})$. Для будь-якої внутрішньої точки \bar{x} області визначення M множина $G_f(\bar{x})$ є непорожньою, обмеженою, опуклою і замкненою (доведення див. в [1]).

Аби внутрішня точка області визначення $f(x)$ $x^* \in M$ була точкою мінімуму, необхідно і достатньо, щоб субградієнтна множина $G_f(x^*)$ містила точку 0.

Геометрично формула (10.10) означає, що *антисубградієнт* в точці \bar{x} утворює гострий кут з довільним напрямком, проведеним з \bar{x} до точки x з меншим значенням $f(x)$. Звідси випливає, що коли X^* є непорожньою, $\bar{x} \notin X^*$, то рух з \bar{x} в напрямку $-g_f(\bar{x})$ з достатньо малим кроком буде зменшувати відстань до X^* . Цей простий факт є основою *субградієнтного методу* чи *методу узагальненого градієнтного спуску* (УГС).

Відомі такі основні джерела, що породжують задачі негладкої оптимізації.

1. *Задачі математичного програмування великої розмірності з блоковою структурою та порівняно невеликим числом зв'язків між блоками.* Використання схем декомпозиції для розв'язання таких задач зводить їх до задач мінімізації (максимізації), як правило, негладких функцій від зв'язуючих змінних або від множників Лагранжа (двійних оцінок), що відповідають зв'язувальним обмеженням.

2. *Задачі мінімізації функції максимуму.* Нехай задано параметричну родину опуклих функцій, визначених на всьому просторі E^n , $\{f_\alpha(x)\}_{\alpha \in A}$. Одним з джерел отримання негладких функцій у опуклому програмуванні є операція взяття поточкового максимуму за параметром α , тобто побудова функції максимуму:

$$F(x) = \sup_{\alpha \in A} f_\alpha(x).$$

Область визначення функції $F(x)$ ($\text{dom}F$) збігається з такими значеннями $x \in E^n$, за яких $\{f_\alpha(x)\}$ є обмеженою зверху за α .

Для кожного $\bar{x} \in \text{dom}F$ визначимо таку підмножину індексів:

$$I(\bar{x}) = \{\alpha \in A: f_\alpha(\bar{x}) = F(\bar{x})\}.$$

Субградієнтна множина $G_F(\bar{x})$ функції F в точці \bar{x} визначається за формулою

$$G_F(\bar{x}) = \overline{\text{conv}}\{\cup_{\alpha \in I(\bar{x})} G_{f_\alpha}(\bar{x})\}, \quad (10.11)$$

де $\overline{\text{conv}}\{M\}$ позначає операцію знаходження мінімальної опуклої замкненої множини, що містить M , $G_{f_\alpha}(\bar{x})$ – субградієнтні підмножини функцій f_α в точці \bar{x} , $\alpha \in I(\bar{x})$.

У випадку, коли $I(\bar{x})$ – скінченна множина, всі крайні точки множини $G_F(\bar{x})$ є субградієнтами деяких функцій f_α , $\alpha \in I(\bar{x})$, в точці \bar{x} , і $G_F(\bar{x})$ є опуклим багатогранником відповідної розмірності.

3. *Лагранжеві оцінки у задачах математичного програмування.* Розглянемо загальну задачу математичного програмування, обмеження якої

розбито на дві групи, одна включає ті, що мають вигляд умови належності, тобто $x \in X \subseteq E^n$, а інша – ті, що складають систему рівнянь.

Отже, маємо задачу такого вигляду:

знайти

$$f^* = \inf_{x \in X} f(x), \quad X \subseteq E^n, \quad (10.12)$$

за обмежень

$$g_i(x) = 0, \quad i = 1, \dots, m. \quad (10.13)$$

Вважатимемо, що X – замкнута підмножина n -вимірного евклідового простору, $g_i(x)$ – неперервні функції, визначені на X .

Для оцінки f^* введемо функцію Лагранжа, а саме:

$$L(x, u) = f(x) + \sum_{i=1}^m u_i g_i(x),$$

де $u = \{u_1, \dots, u_m\}$ – вектор множників Лагранжа.

Розглянемо оцінку

$$\psi(u) = \inf_{x \in X} L(x, u).$$

За будь-якого допустимого x і довільного u маємо $\psi(u) \leq f(x)$, звідки випливає, що $\psi(u) \leq f^*$.

Задача пошуку найкращої оцінки для оптимального значення задачі (10.12), (10.13) у даному класі лагранжевих оцінок зводиться до розв'язання координувальної задачі: знайти

$$\psi^* = \sup_{u \in R^m} \psi(u).$$

Функція $\psi(u)$ є увігнутою як результат операції мінімізації за $x \in X$ параметричної (u – параметр) сім'ї лінійних за u функцій. Будемо вважати, що $\psi(u)$ – увігнута функція з непорожньою областю $\text{dom} \psi$, яка має внутрішні точки. Нехай \bar{x} – точка з $\text{dom} \psi$, тобто $\bar{x} \in \text{int} \text{dom} \psi$.

Тоді за правилами обчислення субградієнта від функції максимуму $\phi(u) = -\psi(u)$ субградієнтна множина $G_\phi(\bar{u})$ визначається у такий спосіб:

$$G_\phi(\bar{u}) = \overline{\text{conv}} \left\{ \bigcup_{x \in X(\bar{u})} G(x(\bar{u})) \right\},$$

де $X(u)$ – множина всіляких розв'язків локальної задачі $\inf_{x \in X} L(x, \bar{u})$; $G(x(\bar{u}))$ – вектор відхилу, який відповідає розв'язку $x(\bar{u})$, $G(x(\bar{u})) = \{g_1(x(\bar{u})), \dots, g_m(x(\bar{u}))\}$. Отже, якщо в точці \bar{x} локальна задача має неєдиний розв'язок, то градієнт функції $\phi(u)$ у відповідній точці є розривним.

4. *Задачі мінімізації функції максимуму, характерні для моделей ігрового характеру, багатокритеріальних моделей оптимального планування та дослідження операцій.* До таких задач зводяться розв'язання систем рівнянь і нерівностей, визначення коефіцієнтів нелінійної регресії, коли

використовується чебишевський критерій мінімізації максимуму відхилу (модуля відхилу).

5. *Задачі нелінійного програмування, для розв'язання яких застосовується метод негладких штрафних функцій.* Негладкі штрафні функції певного вигляду (див. далі) володіють безперечною перевагою порівняно із звичайно застосовуваними гладкими функціями штрафу: при використанні негладких штрафних функцій немає необхідності спрямовувати штрафні коефіцієнти до $+\infty$.

6. *Задачі оптимального управління з неперервним та дискретним часом.* Використання принципу максимуму Понтрягіна або дискретного принципу максимуму у багатьох випадках спричиняє розв'язання задач мінімізації функцій з розривним градієнтом. Їх можна розглядати як спеціальні задачі нелінійного програмування, для розв'язання яких застосовні схеми декомпозиції чи метод негладких штрафних функцій.

7. *Задачі дискретного програмування або задачі змішаного дискретно-неперервного типу.* Багато задач такого ґатунку досить успішно можуть вирішуватися з використанням методу гілок та меж з одержанням оцінок шляхом розв'язання двоїстої задачі. Остання зазвичай виявляється задачею мінімізації опуклої частково-лінійної функції з величезним числом "кусків" при простих обмеженнях, тобто задачею негладкої оптимізації.

8. *Функції з розривним градієнтом як результат частково-гладкої апроксимації техніко-економічних характеристик реальних об'єктів можуть безпосередньо входити до моделі задач оптимального планування, проектування чи дослідження операцій.*

Слід також відзначити, що з прикладної точки зору немає різкої межі між гладкими і негладкими функціями: функція з дуже швидко мінливим градієнтом близька за своїми властивостями до негладкої функції. Тому обчислювальні методи, розроблені для розв'язання задач негладкої оптимізації, виявляються ефективними й для оптимізації "поганих" гладких функцій (функцій яружного типу).

4.2. Узагальнений градієнтний спуск

Методом узагальненого градієнтного спуску (УГС) або субградієнтним методом називається процедура побудови мінімізаційної послідовності $\{x_k\}_{k=0}^{\infty}$, коли її елемент x_k обчислюється за таким правилом:

$$x_{k+1} = x_k - h_k \frac{g_f(x_k)}{\|g_f(x_k)\|}, \quad k = 0, 1, 2, \dots; \quad (10.14)$$

де x_0 – початкове наближення; $g_f(x_k)$ – довільний субградієнт функції $f(x)$ в точці x_k , $h_k > 0$ – кроковий множник. Якщо $g_f(x_{k^*}) = 0$, то x_{k^*} – точка мінімуму функції $f(x)$, і процес (10.14) зупиняється.

Конкретний варіант методу УГС типу (10.14) визначається процедурою вибору величини кроку h_k . Серед відомих способів визначення h_k можна виділити наступні:

- 1) $h_{k+1} = \frac{h_0}{k+1}$, $h_0 > 0$, $k = 0, 1, 2, \dots$;
- 2) $h_{k+1} = \frac{h_k}{k+1}$, $h_0 > 0$, $k = 0, 1, 2, \dots$;
- 3) $h_{k+1} = \frac{h_k + h_{k-1}}{2}$, $h_0 > 0$, $h_1 = \frac{h_0}{2}$, $k = 0, 1, 2, \dots$

Найбільш загальний результат збіжності субградієнтного методу пов'язаний з класичними умовами регулювання кроку і сформульований у наступній теоремі.

Теорема 10.5. Нехай $f(x)$ – опукла функція, визначена на E_n , з обмеженою множиною мінімумів X^* , $\{h_k\}$ ($k = 0, 1, 2, \dots$) – послідовність чисел з такими властивостями:

$$h_k > 0; \quad \lim_{k \rightarrow \infty} h_k = 0; \quad \sum_{k=0}^{\infty} h_k = +\infty.$$

Тоді послідовність $\{x_k\}$ ($k = 1, 2, \dots$), утворена за формулою (10.14), при довільному $x_0 \in E_n$ має одну з таких властивостей: або знайдеться такий номер $k = \bar{k}$, при якому $x_{\bar{k}} \in X^*$, або $\lim_{k \rightarrow \infty} \min_{y \in X^*} \|x_k - y\| = 0$, $\lim_{k \rightarrow \infty} f(x_k) = \min_{y \in E^n} f(x) = f^*$.

Є кілька варіантів доведення теореми 10.5 та її аналога для УГС без нормування субградієнта:

$$x_{k+1} = x_k - h_k g_f(x_k), \quad k = 0, 1, 2, \dots$$

Усі вони засновані на вивченні поведінки послідовності $\{\rho_k\}_{k=0}^{\infty}$, де $\rho_k = \min_{x \in X^*} \|x_k - x\|$.

Лема 10.1. Нехай в субградієнтному методі (10.14) використовується регулювання кроку $h_k = h$, $h > 0$, для всіх $k = 0, 1, 2, \dots$. Тоді для будь-якого $\varepsilon > 0$ і $x^* \in X^*$ знайдуться номер $k = k^*$ і елемент $x \in E_n$, які будуть задовольняти наступні умови:

$$f(\bar{x}) = f(x_{k^*}), \quad \text{причому } \|\bar{x} - x^*\| < \frac{h}{2}(1 + \varepsilon).$$

Лема 10.1 гарантує зменшення відстані до точок області мінімумів тільки у випадку досить великої відстані від множини мінімумів до точок x_k . Однак за допомогою цієї леми можна обґрунтувати збіжність субградієнтного методу за додаткових припущень про властивості мінімізованої функції. Так, наприклад, якщо множина точок мінімуму містить сферу радіуса $r > h/2$, то субградієнтний метод з сталим кроком h гарантує знаходження такого значення k^* , що $x_{k^*} \in X^*$.

За певних додаткових припущень деякі варіанти УГС збігаються зі швидкістю геометричної прогресії.

Теорема 10.6. Нехай $f(x)$ – опукла функція, визначена на E_n , і для всіх $x \in E_n$ при деякому ϕ ($0 \leq \phi \leq \pi/2$) виконується нерівність

$$(g_f(x), x - x^*(x)) \geq \cos \phi \|g_f(x)\| \cdot \|x - x^*(x)\|, \quad (10.15)$$

де $x^*(x)$ – точка, що належить множині точок мінімуму функції $f(x)$ і перебуває на найкоротшій відстані від x . Тоді, якщо за будь-якого x_0 обрати величину h_0 , що задовольняє нерівність

$$h_0 \geq \begin{cases} \|x^*(x_0) - x_0\| \cos \phi, & \pi/4 \leq \phi \leq \pi/2, \\ \|x^*(x_0) - x_0\| / (2 \cos \phi), & 0 \leq \phi < \pi/4, \end{cases}$$

визначити $\{h_k\}_{k=0}^{\infty}$ за рекурентною формулою

$$h_{k+1} = h_k r(\phi), \quad k = 0, 1, 2, \dots,$$

де

$$r(\phi) = \begin{cases} \sin \phi, & \pi/4 \leq \phi < \pi/2, \\ \frac{1}{2 \cos \phi}, & 0 \leq \phi < \pi/4, \end{cases}$$

і обчислити $\{x_k\}_{k=1}^{\infty}$ за формулою (10.14), то або для деякого індексу k^* $g_f(x_{k^*}) = 0$ і x_{k^*} належить множині мінімумів, або для всіх $k = 0, 1, 2, \dots$ виконується нерівність

$$\|x_k - x^*(x_k)\| \leq \begin{cases} h_{k+1} / \cos \phi, & \pi/4 \leq \phi < \pi/2, \\ 2 \cos \phi \cdot h_{k+1}, & 0 \leq \phi < \pi/4. \end{cases}$$

Таким чином, якщо кут ϕ заздалегідь відомий, то, регулюючи крок за формулами теореми 10.6, можна отримати збіжність ітераційного процесу (10.14) до точки мінімуму зі швидкістю геометричної прогресії зі знаменником $q = r(\phi)$. В (10.14) величина $\cos \phi$ характеризує ступінь витягнутості поверхонь рівня функції $f(x)$.

Якщо в деякому околі мінімуму функції $f(x)$ не існує такого кута $\phi < \pi/2$, що для будь-якого x з цього околу виконується (10.15), то таку функцію називають *суттєво яружною*. Під час мінімізації істотно яружних функцій наведений у теоремі 10.6 спосіб регулювання крокових множників непридатний. У цьому випадку слід використовувати універсальний спосіб вибору крокових множників (див. теорему 10.5).

Наступна теорема, аналогічна до теореми 10.6, сформульована безпосередньо в термінах, що характеризують ступінь "витягнутості" поверхонь рівня.

Теорема 10.7. Припустимо опукла функція $f(x)$ визначена на всьому просторі E_n , x^* – одна єдина точка мінімуму $f(x)$ і задані початкове наближення x_0 та числа σ і h_0 , причому $\sigma \geq \sqrt{2}$, $h_0 \geq \|x_0 - x^*\| / \sigma$. Нехай $Y = \{y: \|y - x^*\| \leq \sigma h_0\}$. Якщо для будь-якої пари точок $x, z \in Y$, при якій $f(x) = f(z) \neq f(x^*)$, виконується умова

$$\|x - x^*\| \leq \sigma \|z - x^*\|,$$

то послідовність $\{x_k\}_{k=0}^{\infty}$, отримана за рекурентною формулою (10.14), де $h_{k+1} = h_k \sqrt{\sigma^2 - 1} / \sigma$, прямує до x^* зі швидкістю геометричної прогресії:

$$\|x - x^*\| \leq h_k \sigma,$$

за виключенням випадку, коли для деякого $k = \bar{k}$ $g_f(x_{\bar{k}}) = 0$, тобто $x_{\bar{k}} = x^*$.

Ще одним варіантом узагальненого градієнтного спуску, який збігається зі швидкістю геометричної прогресії, є субградієнтний метод, де кроковий множник залишається протягом певного числа кроків сталим, а потім зменшується вдвічі [1].

Теорема 10.8. Припустимо опукла функція $f(x)$ визначена на всьому просторі E_n , x^* – одна єдина точка мінімуму $f(x)$ і задані початкове наближення x_0 і числа σ та h_0 , причому $\sigma \geq 2$, $h_0 \geq \|x_0 - x^*\|/\sigma$. Нехай

$$Y = \{y: \|y - x^*\| \leq \sigma h_0\}.$$

Якщо для будь-якої пари точок $x, z \in Y$, при якій $f(x) = f(z) \neq f(x^*)$, виконується умова

$$\|x - z\| \leq \sigma,$$

то для послідовності точок $\{x_k\}_{k=0}^{\infty}$, побудованих за формулою (10.14), коли $h_k = h_0 \cdot 2^{-\lfloor k \rfloor / N}$, де $\lfloor a \rfloor$ – ціла частина числа a , при достатньо великому h_0 і $N \geq 3\sigma^2 + 1$ виконується нерівність

$$\|x_k - x^*\| \leq 2\sigma h_k, \quad k = 0, 1, 2, \dots$$

Теорема 10.8 сформульована в термінах, що характеризують ступінь витягнутості поверхонь рівня, і легко бачити, що чим більше σ , тим меншою буде швидкість збіжності субградієнтного методу.

Методи УГС дозволяють вирішувати завдання виробничо-транспортного планування із застосуванням схем декомпозиції для задач великої розмірності.

Задачі для розв'язування у класі

1. Розв'язати задачу оптимізації за допомогою методів найшвидшого спуску (підйому) для функції $z(x, y) = x^2 - 6xy + 2y^2 - 4x + y - 2$ градієнтним методом з дробленням кроку, з апіорним вибором кроку. Надати геометричну інтерпретацію процесу розв'язання.

2. З початкового наближення $(x^0, y^0) = (-2, 1)$ здійснити два кроки методу найшвидшого спуску для задачі мінімізації такої функції:

$$z(x, y) = x^2 + xy + 2y^2 + x.$$

3. З початкового наближення $(x^0, y^0) = (1, -1)$ здійснити один крок градієнтного методу з вибором крокового множника за правилом Армїхо для задачі мінімізації функції $z(x, y) = 2x^2 + xy + 3y^2 + 5$. Обрати такі параметри методу: $\hat{\alpha} = 1$, $\varepsilon = \theta = 1/2$.

Рекомендована література

1. Стецюк П.І. Методи еліпсоїдів та g -алгоритми. // Кишиневу, Еврика, 2014. – 488 с.
2. Вітлінський В.В. Економіко-математичні методи та моделі: оптимізація : навч. посібник [Електронний ресурс] / В. В. Вітлінський, Т. О. Терещенко, С. С. Савіна – К. : КНЕУ, 2016. – 303 с.

3. Жалдак М.І., Триус Ю.В. Основи теорії і методів оптимізації: Навчальний посібник. – Черкаси: Брама-Україна, 2005. – 608 с.
4. Сікора Я. Б. Методи оптимізації та дослідження операцій [Текст] : навчальний посібник / Я. Б. Сікора, А.Й. Щехорський, Б.Л. Якимчук. – Житомир: Вид-во ЖДУ ім. Івана Франка, 2019. – 148 с.
5. Кононюк А.Е. Основы теории оптимизации. Безусловная оптимизация К.2.ч.1. Киев: "Освіта України", 2011. – 544 с.
6. Кіселева О.М. Неперервні задачі оптимального розбиття множин та г-алгоритми: Монографія / О.М. Кіселева, Л.С. Коряшкіна.– К.: Наук. думка, 2015. – 400 с.

Контрольні питання

1. Дайте визначення напрямку спуску.
2. Які методи багатовимірної мінімізації функцій є методами спуску?
3. Запишіть умову одновимірного пошуку в багатовимірному просторі.
4. Наведіть формули ітерації методу спуску.
5. Складіть алгоритм методу спуску.
6. Як виконується масштабування напрямку пошуку в методі спуску?
7. Дайте визначення напрямку найшвидшого спуску.
8. Опишіть спосіб найшвидшого спуску.
9. Як обчислюється величина кроку у методі найшвидшого спуску?
10. Яка ідея способу Арміхо одновимірного пошуку параметра довжини кроку в методах спуску?
11. Як закінчуються обчислення в методі найшвидшого спуску?
12. Якого порядку метод найшвидшого спуску?
13. Складіть алгоритм методу найшвидшого спуску.
14. Яка особливість траєкторії пошуку для методу найшвидшого спуску?
15. Вкажіть переваги та недоліки методу найшвидшого спуску.
16. Навіщо потрібно вміти обчислювати градієнт функції багатьох змінних?
17. Вкажіть способи обчислення градієнта функції багатьох змінних.
18. Запишіть формулу кінцевої різниці вперед для обчислення градієнта функції багатьох змінних.
19. Яка похибка формули кінцевої різниці «вперед» для обчислення градієнта функції багатьох змінних?

Завдання для самостійної роботи

Знайти екстремум функції $z(x, y) = ax^2 + bxy + cy^2 + dx + ey + f$, використовуючи а) градієнтний метод з дробленням кроку; б) градієнтний метод з апіорним вибором кроку; в) метод найшвидшого спуску (підйому). Методи реалізувати у вигляді комп'ютерної програми будь-якою мовою програмування. Передбачити довільне задання параметрів функції. У випадку, коли функція необмежена зверху або знизу, видавати відповідне повідомлення. Здійснити порівняння методів стосовно швидкості пошуку екстремуму.

Практичне заняття № 11

Методи другого порядку. Метод Ньютона і його модифікації

Мета: ознайомитися з методом Ньютона і його модифікаціями; навчитися розв'язувати задачу багатовимірної безумовної оптимізації методом Ньютона

План заняття

1. Метод Ньютона і його модифікації
2. Збіжність методу Ньютона і оцінка швидкості збіжності
3. Метод Ньютона з регулюванням кроку

Опорні поняття

1. Метод Ньютона і його модифікації

Це метод безумовної багатовимірної оптимізації другого порядку, який використовує обчислення других частинних похідних цільової функції. Для з'ясування, чи є дана точка мінімумом, максимумом або сідлом, доводиться досліджувати другі похідні функції. В модифікаціях методу (квазіньютонівських алгоритмах) матриця других частинних похідних апроксимується за допомогою інформації про значення градієнтів функції, і, відтак, ці модифікації є методами першого порядку.

Перейдемо до опису методу Ньютона. Вважатимемо, що в задачі

$$f(x) \rightarrow \min_{x \in E^n} \quad (11.1)$$

функція $f(x) \in C^2(E^n)$, причому матриця $f''(x)$ є невиродженою на E^n .

Послідовність $\{x^k\}_{k=0}^{\infty}$ генерується з наступних міркувань. Оскільки цільова функція є двічі неперервно диференційовною, для поточної точки x^k можемо записати таку рівність:

$$f(x) - f(x^k) = (f'(x^k), x - x^k) + \frac{1}{2}(f''(x^k)(x - x^k), x - x^k) + o(\|x - x^k\|^2).$$

Для визначення наступної $(k + 1)$ -ї точки мінімізується функція $f_k(x)$, яка є квадратичною частиною приросту $f(x) - f(x^k)$, тобто розв'язується наступна задача:

$$f_k(x) \rightarrow \min_{x \in E^n}, \quad (11.2)$$

де

$$f_k(x) = (f'(x^k), x - x^k) + \frac{1}{2}(f''(x^k)(x - x^k), x - x^k).$$

Виходячи із диференціальних умов оптимальності маємо, що в точці мінімуму функції $f_k(x)$ її градієнт $f_k'(x)$ перетворюється на нуль, тобто

$$f_k^{(x)} = f'(x^k) + f''(x^k)(x - x^k) = 0. \quad (11.3)$$

Якщо матриця Гессе $f''(x^k)$ системи лінійних алгебраїчних рівнянь (11.3) не вироджена, то, розв'язуючи рівняння (11.3) отримаємо точку

$$x^{k+1} = x^k + h^k, \quad h^k = -\left(f''(x^k)\right)^{-1} f'(x^k). \quad (11.4)$$

Вочевидь, $f_k''(x) = f''(x^k)$. Якщо, наприклад, вихідна функція $f(x)$, яка мінімізується, є опуклою на E^n , то опуклою є і функції $f_k(x)$. Дійсно, необхідною і достатньою умовами опуклості функції є невід'ємна визначеність матриці її других похідних, а функція $f(x)$ опукла згідно з умовою, тоді і функція $f_k(x)$ є опуклою. При цих умовах точка x^{k+1} з (11.4) є точкою мінімуму квадратичної апроксимації $f_k(x)$. Точку x^{k+1} , яка отримана за формулою (11.4), вважаємо за $(k+1)$ -е наближення методу Ньютона.

Коли ж функція $f(x)$ не є опуклою, точка x^{k+1} з (11.4) є стаціонарною і її необхідно додатково дослідити на оптимальність.

Неважко побачити, що метод Ньютона мінімізації функції $f(x)$, який визначається співвідношенням (11.4), є застосуванням відомого з курсу методів обчислень методу Ньютона розв'язання системи нелінійних рівнянь

$$f'(x) = 0.$$

2. Збіжність методу Ньютона і оцінка швидкості збіжності

Теорема 11.1. Припустимо: 1) функція $f(x)$ двічі неперервно-диференційовна на E^n ; 2) $f(x)$ сильно опукла з константою $\theta > 0$ на E^n , тобто має місце співвідношення

$$f(\mu x + (1 - \mu)y) \leq \mu f(x) + (1 - \mu)f(y) - \theta \mu(1 - \mu)\|x - y\|^2 \quad \forall x, y \in E^n; \quad (11.5)$$

3) матриця її других похідних задовольняє умову Ліпшиця, а саме:

$$\|f''(x) - f''(y)\| \leq M\|x - y\| \quad \forall x, y \in E^n. \quad (11.6)$$

де $M > 0$ – константа Ліпшиця;

4) початкове наближення x^0 таке, при якому

$$\|f'(x^0)\| = \frac{8\theta^2 q}{M}, \quad (11.7)$$

де $q \in (0,1)$.

Тоді послідовність (11.4) збігається до точки мінімуму x^* з квадратичною швидкістю і має місце така оцінка:

$$\|x^k - x^*\| \leq \frac{4\theta q^{2^k}}{M}.$$

Отже, збіжність методу Ньютона обґрунтовано лише для достатньо близького до точки мінімуму початкового наближення x^0 . При цьому умову (11.7), яка гарантує збіжність методу для даного початкового наближення, на практиці важко перевірити, оскільки константи M і θ , як правило, невідомі заздалегідь. Відтак, залежність збіжності методу від початкового наближення є недоліком методу Ньютона.

Метод також характеризується високою трудомісткістю, яка обумовлена необхідністю обчислення і обертання на кожному кроці матриці Гессе. У зв'язку з цими двома недоліками застосування класичного методу Ньютона далеко не завжди дає успіх. Модифікації його спрямовані на те, щоб, зберігаючи основну його перевагу – швидку збіжність, зменшити трудомісткість та ослабити вимоги до вибору початкового наближення.

3. Метод Ньютона з регулюванням кроку

Ця модифікація описується наступною формулою.

$$x^{k+1} = x^k + \alpha_k h^k, \quad h^k = -\left(f''(x^k)\right)^{-1} f'(x^k). \quad (11.8)$$

Коли $\alpha_k = 1$ вона співпадає з класичним методом Ньютона. Вибір коефіцієнтів $\alpha_k > 0$ в (11.8) здійснюється, зазвичай, з умови мінімізації цільової функції вздовж заданого напрямку $h^k = -\left(f''(x^k)\right)^{-1} f'(x^k)$, або за допомогою процедури дроблення кроку, яка задовольняє умови релаксації ітераційного процесу.

Можна показати, що вказані варіанти методу (11.8) збігаються для будь-якого початкового наближення $x^0 \in E^n$. При цьому швидкість збіжності буде або надлінійною, або квадратичною залежно від вимог, яким буде задовольняти цільова функція $f(x)$. Зменшити трудомісткість методу можна, якщо обчислювати матрицю Гессе не на кожному кроці, як це робиться в (11.8), а один раз через кожні s кроків, тобто

$$x^{k+1} = x^k + \alpha_k h^k, \quad h^k = -\left(f''\left(x^{s\left[\frac{k}{s}\right]}\right)\right)^{-1} f'(x^k), \quad \alpha_k > 0.$$

де $\left[\frac{k}{s}\right]$ – ціла частина числа $\frac{k}{s}$, тобто найбільше ціле число, яке менше або дорівнює $\frac{k}{s}$. Підбираючи емпіричним шляхом s , іноді вдається отримати за допомогою цього методу непогані результати. Однак, істотного зменшення трудомісткості методу Ньютона цей прийом не дає.

Більш перспективним у цьому плані є інший підхід, при якому будується апроксимація матриці $\left(f''(x^k)\right)^{-1}$ на основі інформації про значення градієнтів $f'(x^k), f'(x^{k-1}), \dots$. Такі методи називаються *квазіньютонівськими* і будуть розглянуті далі.

Приклад 11.1. застосуємо метод Ньютона для задачі мінімізації функції $f(x) = 5x_1^2 + 4x_2^2 + 3x_1x_2 - 27x_1 - 18x_2 + 115$.

Розв'язання. Нехай $x^0 = \begin{pmatrix} -1 \\ 2 \end{pmatrix}$. Обчислимо градієнт і матрицю других частинних похідних в цій точці, тобто

$$f'(x) = \begin{pmatrix} 10x_1 + 3x_2 - 27 \\ 3x_1 + 8x_2 - 18 \end{pmatrix};$$

$$f'(x^0) = \begin{pmatrix} -10 + 6 - 27 \\ -3 + 16 - 18 \end{pmatrix} = \begin{pmatrix} -31 \\ -5 \end{pmatrix};$$

$$f''(x) = \begin{pmatrix} 10 & 3 \\ 3 & 8 \end{pmatrix}; (f''(x^0))^{-1} = \begin{pmatrix} \frac{8}{71} & -\frac{3}{71} \\ -\frac{3}{71} & \frac{10}{71} \end{pmatrix}.$$

Знайдемо наступне наближення за формулою методу Ньютона:

$$x^1 = x^0 - (f''(x^0))^{-1} f'(x^0) = \begin{pmatrix} -1 \\ 2 \end{pmatrix} - \begin{pmatrix} \frac{8}{71} & -\frac{3}{71} \\ -\frac{3}{71} & \frac{10}{71} \end{pmatrix} \begin{pmatrix} -31 \\ -5 \end{pmatrix} = \begin{pmatrix} 2,28169 \\ 1,39437 \end{pmatrix}.$$

Оскільки $f'(x^1) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$, робимо висновок, що за одну ітерацію методу отримали стаціонарну точку. З огляду на додатну визначеність матриці Гессе заключаємо, що знайдена $x^1 = \begin{pmatrix} 2,28169 \\ 1,39437 \end{pmatrix}$ є точкою мінімуму цільової функції.

Отже, для квадратичної функції метод Ньютона збігається за одну ітерацію.

Приклад 11.2. Розглянемо задачу пошуку екстремуму функції

$$f(x) = (x_2 - x_1^2)^2 + 9(x_1 - x_2)^2.$$

Розв'язання. Нехай $x^0 = \begin{pmatrix} 3 \\ 2 \end{pmatrix}$. Обчислення градієнта, матриці других частинних похідних, вектора спуску в поточних наближеннях запишемо у табл. 11.1. Отримана точка мінімуму $x^* = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$.

Результати обчислень в прикладі 2

k	x^k	$f(x^k)$	$f'(x^k)$	$f''(x^k)$	h^k
0	$\begin{pmatrix} 3 \\ 2 \end{pmatrix}$	58,00	$\begin{pmatrix} 102 \\ -16 \end{pmatrix}$	$\begin{pmatrix} 118 & -30 \\ -30 & 20 \end{pmatrix}$	$\begin{pmatrix} -2,07 \\ -0,80 \end{pmatrix}$
1	$\begin{pmatrix} 1,93 \\ 1,2 \end{pmatrix}$	11,27	$\begin{pmatrix} 32,79 \\ -6,54 \end{pmatrix}$	$\begin{pmatrix} 58,0 & -25,7 \\ -25,7 & 20 \end{pmatrix}$	$\begin{pmatrix} -0,98 \\ -0,93 \end{pmatrix}$
2	$\begin{pmatrix} 0,95 \\ 0,26 \end{pmatrix}$	4,67	$\begin{pmatrix} 14,83 \\ -2,66 \end{pmatrix}$	$\begin{pmatrix} 27,8 & -21,8 \\ -21,8 & 20 \end{pmatrix}$	$\begin{pmatrix} -2,95 \\ -3,09 \end{pmatrix}$
3	$\begin{pmatrix} -2 \\ -2,82 \end{pmatrix}$	52,76	$\begin{pmatrix} -39,92 \\ -15,31 \end{pmatrix}$	$\begin{pmatrix} 77,4 & -10 \\ -10 & 20 \end{pmatrix}$	$\begin{pmatrix} 0,66 \\ 1,09 \end{pmatrix}$
4	$\begin{pmatrix} -1,35 \\ -1,73 \end{pmatrix}$	13,86	$\begin{pmatrix} -12,11 \\ -7,85 \end{pmatrix}$	$\begin{pmatrix} 46,6 & -12,6 \\ -12,6 & 20 \end{pmatrix}$	$\begin{pmatrix} 0,44 \\ 0,67 \end{pmatrix}$
5	$\begin{pmatrix} -0,9 \\ -1,06 \end{pmatrix}$	3,74	$\begin{pmatrix} -3,98 \\ -4,06 \end{pmatrix}$	$\begin{pmatrix} 32,0 & -14,4 \\ -14,4 & 20 \end{pmatrix}$	$\begin{pmatrix} 0,32 \\ 0,43 \end{pmatrix}$
6	$\begin{pmatrix} -0,59 \\ -0,63 \end{pmatrix}$	0,96	$\begin{pmatrix} -1,52 \\ -2,02 \end{pmatrix}$	$\begin{pmatrix} 24,6 & -15,7 \\ -15,7 & 20 \end{pmatrix}$	$\begin{pmatrix} 0,25 \\ 0,3 \end{pmatrix}$
7	$\begin{pmatrix} -0,33 \\ -0,33 \end{pmatrix}$	0,19	$\begin{pmatrix} -0,68 \\ -0,87 \end{pmatrix}$	$\begin{pmatrix} 20,7 & -16,7 \\ -16,7 & 20 \end{pmatrix}$	$\begin{pmatrix} 0,21 \\ 0,22 \end{pmatrix}$
8	$\begin{pmatrix} -0,13 \\ -0,11 \end{pmatrix}$	0,02	$\begin{pmatrix} -0,32 \\ -0,23 \end{pmatrix}$	$\begin{pmatrix} 18,6 & -17,5 \\ -17,5 & 20 \end{pmatrix}$	$\begin{pmatrix} 0,15 \\ 0,15 \end{pmatrix}$
9	$\begin{pmatrix} 0,03 \\ 0,03 \end{pmatrix}$	0,00	$\begin{pmatrix} -0,11 \\ 0,08 \end{pmatrix}$	$\begin{pmatrix} 17,9 & -18,1 \\ -18,1 & 20 \end{pmatrix}$	$\begin{pmatrix} 0,03 \\ 0,03 \end{pmatrix}$
10	$\begin{pmatrix} 0,06 \\ 0,06 \end{pmatrix}$	0,00	$\begin{pmatrix} 0,00 \\ 0,01 \end{pmatrix}$	$\begin{pmatrix} 178 & -18,2 \\ -18,2 & 20 \end{pmatrix}$	$\begin{pmatrix} 0,08 \\ 0,08 \end{pmatrix}$
11	$\begin{pmatrix} -0,02 \\ -0,02 \end{pmatrix}$	0,00

Задачі для розв'язування в класі

1. За допомогою методу Ньютона розв'язати задачу мінімізації функції

$$z(x, y) = x^2 - 6xy + 2y^2 - 4x + y - 2.$$

2. З початкового наближення $(x^0, y^0) = (-2, 1)$ здійснити два кроки методу Ньютона з регулюванням кроку для задачі мінімізації такої функції:

$$z(x, y) = 10x^4 + xy^2 - 2x^3y + 2y^4 + 7x.$$

3. З початкового наближення $(x^0, y^0) = (1, -1)$ здійснити один крок методу Ньютона з вибором кроку за правилом Арміхо, в якому $\hat{\alpha} = 1, \varepsilon = \theta = 1/2$ для задачі мінімізації функції $z(x, y) = 2x^2 + xy^2 - y + 3y^4 + 5$.

Рекомендована література і інформаційні ресурси

1. Стецюк П.І. Методи еліпсоїдів та g -алгоритми. // Кишиневу, Еврика, 2014. – 488 с.
2. Вітлінський В.В. Економіко-математичні методи та моделі: оптимізація : навч. посібник [Електронний ресурс] / В. В. Вітлінський, Т. О. Терещенко, С. С. Савіна – К. : КНЕУ, 2016. – 303 с.
3. Жалдак М.І., Триус Ю.В. Основи теорії і методів оптимізації: Навчальний посібник. – Черкаси: Брама-Україна, 2005. – 608 с.
4. Сікора Я. Б. Методи оптимізації та дослідження операцій [Текст] : навчальний посібник / Я. Б. Сікора, А.Й. Щехорський, Б.Л. Якимчук. – Житомир: Вид-во ЖДУ ім. Івана Франка, 2019. – 148 с.
5. Кононюк А.Е. Основы теории оптимизации. Безусловная оптимизация К.2.ч.1. Киев: "Освіта України", 2011. – 544 с.
6. Кіселева О.М. Неперервні задачі оптимального розбиття множин та g -алгоритми: Монографія / О.М. Кіселева, Л.С. Коряшкіна.– К.: Наук. думка, 2015. – 400 с.

Контрольні питання

1. Чим визначається порядок методу оптимізації? Який порядок має метод Ньютона?
2. Яким чином задається вектор спуску в методі Ньютона?
3. Що потрібно змінити в ітераційній формулі методу Ньютона, якщо розв'язувати задачу пошуку максимуму цільової функції?
4. Назвіть переваги і недоліки методу Ньютона.
5. Який недолік методу Ньютона усуває його модифікація з регулюванням кроку?
6. Який недолік методу Ньютона усуває його модифікація з оновленням матриці других частинних похідних?
7. Назвіть умови збіжності методу Ньютона. Якою буде швидкість збіжності методу за цих умов?
8. Опишіть способи наближеного обчислення градієнта функції, гессіана?
9. Які критерії зупинки ітераційного процесу застосовуються у реалізації методу Ньютона для задачі багатовимірної безумовної оптимізації?
10. Яким умовам має задовольняти цільова функція, аби для її мінімізації можна було застосувати метод Ньютона?

Завдання для самостійної та індивідуальної роботи

Реалізувати програмно алгоритм методу Ньютона і будь-якої його модифікації, використовуючи програмне середовище за своїм вибором, і знайти мінімум функції відповідно до визначеного варіанта.

Варіанти завдань

1. $f(x) = x_1^4 + x_2^4 + x_1 - 14x_1x_2 - 2x_2$;
2. $f(x) = (x_2 - x_1^2)^2 + 100(1 - x_1)^2$;
3. $f(x) = 5(x_2 - x_1^2)^2 + (4 - x_1x_2)^2$;
4. $f(x) = x_1^4 + 25x_2^4 + 3x_1x_2 - 106x_1 - 32x_2$;
5. $f(x) = 5x_1^4 + 14x_2^4 + 9x_1x_2 - 27x_1 - 82x_2 + 33$;
6. $f(x) = 4x_1^2 + x_2^4 - x_1x_2 + 8x_1 - 12x_2 + 10$;
7. $f(x) = 81(x_2 + x_1^2)^2 + (12 - x_1)^2$;
8. $f(x) = (3x_2 - 2x_1^2)^2 + 4(2 - x_1x_2)^2$;
9. $f(x) = 5x_1^4 + 9x_2^2 + 2x_1 * x_2 - 16x_1 - 7x_2 + 90$;
10. $f(x) = 16x_1^4 + 25x_2^4 + 13x_1x_2 - 42x_1 - 18x_2 + 25$;
11. $f(x) = x_1^4 + 4x_2^2 + x_3^2 + x_1 - 2x_2 - 7x_3 + 43$;
12. $f(x) = 4(x_2 - x_1^2)^2 + 64(1 - x_2)^2$;
13. $f(x) = (x_2 - x_1^2)^2 + 49(1 - x_1x_2)^2$;
14. $f(x) = 10x_1^2 + x_2^4 + 4x_1x_2 - 28x_1^2 + 32x_2 + 66$;
15. $f(x) = x_1^4 + 14x_2^2 - 5x_1x_2 + 17x_1^2 - 2x_2 + 3$;
16. $f(x) = 3x_1^4 + 24x_2^4 + x_3^2 + 8x_1^2x_2 - 7x_2 - 12x_3 + 35$;
17. $f(x) = 81(x_2 - x_1^2)^2 + (6 - x_2)^2$;
18. $f(x) = (x_1 - x_2^2)^2 + (9 - x_1x_2)^2$;
19. $f(x) = 25x_1^4 + 64x_2^2 + 14x_1x_2 - 16x_1^2 - 45x_2 + 100$;
20. $f(x) = 9x_1^2 + 16x_2^2 - 2x_1x_2 - 17x_1 - 18x_2 + 19$;
21. $f(x) = x_1^2 + 49x_2^4 + 5x_1x_2 - 8x_1 + 90x_2 + 83$;
22. $f(x) = 100(x_2 - x_1^2)^2 + (4 + x_2)^2$;
23. $f(x) = (x_2 - x_1^2)^2 + 36(2 - x_1x_2)^2$;
24. $f(x) = 9x_1^4 + 9x_2^4 + 14x_1x_2 - 35x_1 - 82x_2 + 7x_1^2$;
25. $f(x) = 5x_1^4 + 14x_2^2 - 9x_1x_2 - 27x_1 - 55x_2 + 59$;
26. $f(x) = 4x_1^2 + 49x_2^4 + 11x_1x_2^2 + 12x_1 - 44x_2 + 69$;
27. $f(x) = 16(x_1 + x_2^2)^2 + (4x_1 + x_2)^2$;
28. $f(x) = 49(x_2 - x_1^2)^2 + (25 + x_1x_2)^2$;
29. $f(x) = 9x_1^4 + 25x_2^2 + 12x_1x_2 - 72x_1 + 35x_2 + 235$;
30. $f(x) = 15x_1^2 + 24x_2^4 + 13x_1x_2 - 47x_1 - 8x_2^2 + 5$.

Практичне заняття № 12

Методи перемінної метрики. Квазіньютонівські методи

Мета: ознайомитися з методами безумовної мінімізації функцій багатьох змінних, які відносяться до методів першого порядку, а за швидкістю збіжності близькі до методу Ньютона

План заняття

1. Постановка задачі безумовної оптимізації;
2. Алгоритм Бroyдена – Флетчера – Гольдфарба – Шанно;
3. BFGS з обмеженою пам'яттю (L-BFGS або LM-BFGS);
4. Метод спряжених градієнтів (метод Флетчера – Рівза);
5. Усічені методи Ньютона;
6. Метод Пауелла;
7. Послідовне квадратичне програмування (SQP).

Опорні поняття

1. Постановка задачі

В цьому розділі будемо досліджувати задачу оптимізації у такій постановці:

$$\arg \min_x f(x), \quad f(x) \in C^1(R^n).$$

2. Алгоритм Бroyдена – Флетчера – Гольдфарба – Шанно

Алгоритм Бroyдена – Флетчера – Гольдфарба – Шанно (BFGS) – ітераційний метод обчислювальної оптимізації, призначений для знаходження локального максимуму або мінімуму нелінійного функціоналу без обмежень. BFGS – один з найбільш широко застосовуваних *квазіньютонівських методів*. У них не обчислюється безпосередньо гессіан функції, замість цього його оцінюють приблизно, виходячи із зроблених до цього кроків. Даний метод спрямований на пошук мінімуму будь якої опуклої функції, яка є двічі неперервно диференційовною. Незважаючи на ці теоретичні обмеження, як показує досвід, BFGS добре працює і з неопуклими функціями.

У методах другого порядку задача мінімізації функції розв'язується ітераційно, шляхом її розкладання у поліном другого ступеня, а саме:

$$f(x_k + p) = f(x_k) + \nabla f^T(x_k)p + \frac{1}{2}p^T H(x_k)p,$$

де H – гессіан функціоналу f в точці x .

Дуже часто обчислення гессіана є трудомісткою операцією, тому BFGS алгоритм замість справжнього значення $H(x)$ обчислює його наближене значення B_k , після чого знаходить мінімум отриманої квадратичної задачі:

$$p_k = -B_k^{-1} \nabla f(x_k).$$

Як правило, після цього здійснюється пошук вздовж даного напрямку точки, для якої виконуються умови Вульфа.

Як початкове наближення гессіана можна брати будь-яку невідроджену, добре обумовлену матрицю. Часто за нього беруть одиничну матрицю. На наступному кроці наближене значення гессіана обчислюється за такою формулою:

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}.$$

де I – одинична матриця, $s_k = x_{k+1} - x_k$ – крок алгоритму на ітерації k , $y_k = \nabla f_{k+1} - \nabla f_k$ – зміна градієнта та ітерації k .

Оскільки знаходження оберненої матриці є обчислювально складною операцією, замість того, щоб розраховувати B_k^{-1} , обчислюють обернену до B_k матрицю $C_k = B_k^{-1}$:

$$C_{k+1} = (I - \rho_k s_k y_k^T) C_k (I - \rho_k y_k s_k^T) + \rho_k s_k s_k^T,$$

де $\rho_k = \frac{1}{y_k^T s_k}$.

Отже, опишемо алгоритм BFGS .

1. Дано ε, x_0 .
2. Ініціалізуємо $C_0, k = 0$.
3. Доки $\|\nabla f_k\| > \varepsilon$,
 - 3.1. Знаходимо напрямок $p_k = -C_k \nabla f_k$,
 - 3.2. Обчислюємо $x_{k+1} = x_k + \alpha_k p_k$, α_k – задовольняє критеріям Вульфа,
 - 3.3. Позначаємо $s_k = x_{k+1} - x_k$ та $y_k = \nabla f_{k+1} - \nabla f_k$,
 - 3.4. Обчислюємо C_{k+1} ,
 - 3.5. $k = k + 1$.

Алгоритм описано.

Алгоритм BFGS з обмеженою пам'яттю

BFGS з обмеженою пам'яттю (L-BFGS або LM-BFGS) – це алгоритм оптимізації в сімействі квазіньютонівських методів, який наближує алгоритм Бroyдена – Флетчера – Гольдфарба – Шанно (BFGS), використовуючи обмежений об'єм пам'яті комп'ютера. Як і оригінальний BFGS, L-BFGS використовує оцінку зворотної матриці Гессе для керування пошуком у

просторі змінних, але там, де BFGS зберігає наближення розміром $n \times n$ до зворотного гессіана (n – кількість змінних у задачі), L-BFGS зберігає лише частину векторів, які визначають апроксимацію. Завдяки лінійному використанню пам'яті, метод L-BFGS може бути ефективно застосовано для розв'язання задач оптимізації з багатьма змінними. Замість оберненого гессіана H_k , L-BFGS зберігає історію минулих m оновлень положення x та градієнта $\nabla f(x)$, де загалом розмір історії m може бути невеликим.

L-BFGS багато в чому схожий з іншими квазіньютонівськими алгоритмами, але сильно відрізняється тим, як відбувається множення матриці на вектор $d_k = -H_k g_k$, де d_k – приблизний ньютонівський напрямок, g_k – поточний градієнт, а H_k є оберненою матрицею Гессе. Існує декілька опублікованих підходів, що використовують історію оновлень для формування вектор напрямку, а загальноприйнятий підхід має назву "двоциклова рекурсія".

Робота алгоритму починається з того що, для початку беруться x_k , положення на k -й ітерації та $g_k \equiv \nabla f(x_k)$, де f – функція яку мінімізують, а всі вектори є векторами стовпцями. Очікується, що були збережені останні m оновлень у такій формі:

$$s_k = x_{k+1} - x_k,$$

$$y_k = g_{k+1} - g_k.$$

Далі визначається $\rho_k = \frac{1}{y_k^T s_k}$. H_k^0 буде початковим наближенням оберненого гессіана, з якого починається оцінка на ітерації k . Алгоритм базується на рекурсії BFGS для зворотного гессіана, тобто

$$H_{k+1} = (I - \rho_k s_k y_k^T) H_k (I - \rho_k y_k s_k^T) + \rho_k s_k s_k^T.$$

Для фіксованого k визначається послідовність векторів q_{k-m}, \dots, q_k у такий спосіб: $q_k := g_k$ та $q_i := (I - \rho_i y_i s_i^T) q_{i+1}$. Тоді рекурсивний алгоритм для обчислення q_i з q_{i+1} має визначити $\alpha_i := \rho_i s_i^T q_{i+1}$ та $q_i = q_{i+1} - \alpha_i y_i$.

Окрім цього, визначається ще одна послідовність векторів z_{k-m}, \dots, z_k за формулою $z_i := H_i q_i$.

Зауважимо, що існує ще один рекурсивний алгоритм для обчислення цих векторів. В ньому спочатку визначають $z_{k-m} = H_k^0 q_{k-m}$, а потім рекурсивно обчислюють $\beta_i := \rho_i y_i^T z_i$ та $z_{i+1} = z_i + (\alpha_i - \beta_i) s_i$. Значення z_k тоді відповідають напрямку спадання функції.

Отже, обчислити напрямок спуску можна у такий спосіб:

1. $q = g_k$;
2. Для $i = k - 1, k - 2, \dots, k - m$ виконати:
 - 2.1. $\alpha_i = \rho_i s_i^T q$;
 - 2.2. $q = q - \alpha_i y_i$;
3. $\gamma_k = \frac{s_{k-1}^T y_{k-1}}{y_{k-1}^T y_{k-1}}$;

4. $H_k^0 = \gamma_k I$;
5. $z = H_k^0 q$;
6. Для $i = k - m, k - m + 1, \dots, k - 1$ виконати:
 - 6.1. $\beta_i = \rho_i y_i^T z$;
 - 6.2. $z = z + s_i(\alpha_i - \beta_i)$;
7. $z = -z$.

Ця формула дає напрямок пошуку для задачі мінімізації, тобто $z = -H_k g_k$. Також початкова апроксимація зворотнього гессіана H_k^0 вибирається у вигляді діагональної матриці або навіть є кратною одиничній матриці, оскільки це обумовлено обчислювальною ефективністю. Масштабування початкової матриці γ_k гарантує, що напрямок пошуку добре масштабовано, і тому одинична довжина кроку приймається в більшості ітерацій. Лінійний пошук Вулфа використовується, щоб переконатися, що умови кривизни виконано, а оновлення BFGS стабільне.

Метод спряжених градієнтів

Метод спряжених градієнтів (метод Флетчера – Рівза) дозволяє знайти локальний екстремум функції базуючись на інформації про її значення і градієнт. Метод нелінійного спряженого градієнта узагальнює його для застосування у нелінійній оптимізації. У разі квадратичної функції в R^n цей метод дозволяє відшукати мінімум не більше ніж за n кроків. Також він може прямувати за вузькими (погано обумовленими) долинами, де метод найкрутішого спуску сповільнюється і слідує перехресному візерунку.

В задачі на пошук мінімуму функції $f(x)$ з N змінними її градієнт $\nabla_x f$ вказує напрямок максимального збільшення.

Алгоритм починає рух у зворотньому (найкрутішому спуску) напрямку, тобто:

$$\Delta x_0 = -\nabla_x f(x_0)$$

з регульованою довжиною кроку α та, виконуючи лінійний пошук в цьому напрямку, допоки не буде досягнуто мінімуму по f :

$$\alpha_0 := \arg \min_{\alpha} f(x_0 + \alpha \Delta x_0),$$

$$x_1 = x_0 + \alpha_0 \Delta x_0.$$

Після цієї першої ітерації у напрямку найкрутішого спуску Δx_0 наступні кроки становлять одну ітерацію руху по наступному спряженому напрямку s_k , де $s_0 = \Delta x_0$. Таким чином, допоки критерій зупинки не буде задоволено, повторюються наступні кроки:

1. Обчислюється найкрутіший напрямок: $\Delta x_k = -\nabla_x f(x_k)$;
2. Розраховується β_k по одній з відомих формул;
3. Оновлюється спряжений напрямок: $s_k = \Delta x_k + \beta_k s_{k-1}$;

4. Виконується лінійний пошук: $\alpha_k = \arg \min_{\alpha} f(x_k + \alpha s_k)$;

5. Оновлюється позиція: $x_{k+1} = x_k + \alpha_k s_k$;

Найбільш відомими формулами для розрахунку β_k є такі:

- Флетчер – Рівз: $\beta_k^{FR} = \frac{\Delta x_k^T \Delta x_k}{\Delta x_{k-1}^T \Delta x_{k-1}}$;

- Полак – Ріб’єр: $\beta_k^{PR} = \frac{\Delta x_k^T (\Delta x_k - \Delta x_{k-1})}{\Delta x_{k-1}^T \Delta x_{k-1}}$;

- Гестени – Штіфель: $\beta_k^{HS} = \frac{\Delta x_k^T (\Delta x_k - \Delta x_{k-1})}{-s_{k-1}^T (\Delta x_k - \Delta x_{k-1})}$;

- Дай – Юань: $\beta_k^{DY} = \frac{\Delta x_k^T \Delta x_k}{-s_{k-1}^T (\Delta x_k - \Delta x_{k-1})}$.

Усічений метод Ньютона

Усічені методи Ньютона, також відомі як методи оптимізації без гессіана, є сімейством алгоритмів, призначених для оптимізації нелінійних функцій з великою кількістю незалежних змінних. Усічений метод Ньютона полягає у багаторазовому застосуванні ітеративного алгоритму оптимізації для наближеного розв’язання рівнянь Ньютона (тобто виконується лише для обмеженої кількості ітерацій) з метою визначення величин оновлення параметрів функції. З цього випливає, що для роботи усічених методів Ньютона внутрішній розв’язувач має створити хорошу апроксимацію в кінцевій кількості ітерацій.

Як відомо, звичайний метод Ньютона ітеративно будує послідовність точок $(x_k)_{k=0}^{\infty}$ за таким правилом:

$$x_{k+1} = x_k + \alpha_k d_k,$$

де напрямок пошуку d_k знаходиться з наступної системи лінійних рівнянь:

$$\nabla^2 f(x_k) d_k = -\nabla f(x_k),$$

тут $\nabla^2 f(x_k)$ – позитивно визначений гессіан функції f в точці x_k , довжина кроку α_k обирається за допомогою лінійного пошуку.

В усіченому методі Ньютона система розв’язується за допомогою методу сполучених градієнтів. У цьому випадку сама матриця $\nabla^2 f(x_k)$ методу не потрібна, а необхідна тільки процедура множення матриці $\nabla^2 f(x_k)$ на довільний вектор $v \in R^n$. Крім того, не має великого сенсу розв’язувати систему дуже точно, якщо поточна точка x_k знаходиться далеко від оптимуму. Тому, зазвичай, метод сполучених градієнтів зупиняють, як тільки відхил апроксимації $r_k(d) := \nabla^2 f(x_k) d + \nabla f(x_k)$ задовольняє таку умову:

$$\|r_k(d)\|_2 \leq \eta_k \|\nabla f(x_k)\|_2.$$

Послідовність $\eta_k \in (0,1)$ називається *форсувальною* і зазвичай вибирається в такий спосіб:

$$\eta_k := \min \left\{ 0.5, \sqrt{\|\nabla f(x_k)\|_2} \right\}.$$

Це гарантує локальну понадлінійну швидкість збіжності методу в невірродженому випадку.

Метод Пауелла

Метод Пауелла – це алгоритм, запропонований Майклом Дж. Д. Пауеллом для пошуку локального мінімуму функції. Він відноситься до прямих методів (методів нульового порядку), отже, для функції не вимагається диференційованість. Найбільш ефективно за допомогою нього здійснюється мінімізація функцій, близьких до квадратичних.

Алгоритм починає роботу з набору початкових векторів пошуку. Зазвичай береться N пошукових векторів $\{s_1, \dots, s_N\}$, які є просто нормаллями, вирівняними до кожної осі.

Метод мінімізує функцію шляхом двоспрямованого пошуку по кожному з векторів. Його можна здійснити за допомогою методу золотого перетину або методом Брента.

Припустимо, мінімуми, знайдені під час кожного двоспрямованого лінійного пошуку, будуть $\{x_0 + \alpha_1 s_1, x_0 + \sum_{i=1}^2 \alpha_i s_i, \dots, x_0 + \sum_{i=1}^N \alpha_i s_i\}$, де x_0 – початкова вихідна точка, α_i – скаляр, визначений під час двоспрямованого пошуку вздовж s_i . Тоді нове положення точки x_1 можна записати як лінійну комбінацію векторів пошуку, тобто $x_1 = x_0 + \sum_{i=1}^N \alpha_i s_i$.

Новий вектор переміщення $\sum_{i=1}^N \alpha_i s_i$ стає новим вектором пошуку й додається до кінця списку векторів пошуку. Тим часом вектор пошуку, який найбільше сприяв новому напрямку, тобто найбільш успішний $i_d = \arg \max_{i=1}^N |\alpha_i| \|s_i\|$, видаляється зі списку векторів. Новим набором N векторів пошуку буде $\{s_1, \dots, s_{i_d-1}, s_{i_d+1}, \dots, s_N, \sum_{i=1}^N \alpha_i s_i\}$. Алгоритм повторюється довільну кількість разів і зупиняється коли суттєвих покращень не відбувається.

Послідовне квадратичне програмування

Послідовне квадратичне програмування (SQP) – один з найбільш поширених і ефективних оптимізаційних алгоритмів загального призначення, основною ідеєю якого є послідовне розв'язання задач квадратичного програмування, апроксимуючих задачу оптимізації. Для оптимізаційних задач без обмежень алгоритм перетворюється в метод Ньютона пошуку точки, в якій

градієнт цільової функції обертається в нуль. Якщо вихідна задача має обмеження-рівності – то в спеціальну реалізацію ньютонівських методів розв'язання системи Лагранжа.

Базовий алгоритм полягає в тому, що задача нелінійного програмування вирішується ітераційно, починаючи з деякого вектора параметрів x_0 . На $k + 1$ ітерації x_{k+1} буде отримано з x_k за таким правилом:

$$x_{k+1} := x_k + \alpha_k d_k,$$

де d_k – це напрямок пошуку в межах k -го кроку, а α_k – це довжина кроку.

Напрямок пошуку визначають, розв'язуючи підзадачу квадратичного програмування, яка формулюється як квадратична апроксимація функції Лагранжа задачі нелінійного програмування з обмеженнями, а саме:

$$L(x, \lambda) = f(x) - \sum_{j=1}^m \lambda_j g_j(x),$$

та лінійної апроксимації обмежень g_j . Ця задача має стандартну форму квадратичного програмування:

$$\min_{d \in \mathbb{R}^n} \frac{1}{2} d^T B_k d + \nabla f(x_k) d$$

$$\nabla g_j(x_k) d + g_j(x_k) = 0, j = 1, \dots, m_e$$

$$\nabla g_j(x_k) d + g_j(x_k) \geq 0, j = m_e + 1, \dots, m.$$

Такий спосіб визначення напрямку пошуку було запропоновано разом з $B := \nabla_{xx}^2 L(x, \lambda)$. Що стосується способу оновлення матриці B , існують і інші підходи, один з яких бере за ідею відмову від обчислення матриці B_k на кожній ітерації, а замість цього використовує лише інформацію першого порядку для апроксимації гессіана функції Лагранжа за аналогією з квазіньютонівськими методами.

При виборі довжини кроку можна застосовувати наступне правило, для якого було доведено, що одновимірна мінімізація недиференційованої точної штрафної функції

$$\phi(x; \varrho) := f(x) + \sum_{j=1}^{m_e} \varrho_j |g_j(x)| + \sum_{j=m_e+1}^m \varrho_j |g_j(x)|_-$$

разом з $|g_j(x)|_- := \left| \min(0, g_j(x)) \right|$ як функції нагороди $\phi: \mathbb{R}^1 \rightarrow \mathbb{R}^1$

$\phi(\alpha) := \phi(x_k + \alpha d_k)$, і фіксованими x_k та d_k , спричиняє розмір кроку α , який гарантує глобальну збіжність для значень штрафних параметрів ϱ_j , більших ніж деяка нижня межа.

Також відома модифікація методу під назвою *послідовне програмування найменших квадратів* (SLSQP). Вона полягає в заміні підзадачі квадратичного програмування лінійною підзадачею найменших квадратів, використовуючи стабільну LDL^T -факторизацію матриці B , а саме:

$$\min_{d \in \mathbb{R}^n} \left\| (D_k)^{\frac{1}{2}} (L_k)^T d + (D_k)^{-\frac{1}{2}} (L_k)^{-1} \nabla f(x_k) \right\|,$$

$$\nabla g_j(x_k) d + g_j(x_k) = 0, j = 1, \dots, m_e,$$

$$\nabla g_j(x_k) d + g_j(x_k) \geq 0, j = m_e + 1, \dots, m.$$

де $L = [l_1, \dots, l_n]$ – нижня трикутна матриця з одиничними діагональними елементами, $D = \text{diag}(d_1, \dots, d_n)$ – діагональна матриця.

Рекомендована література

1. Нефьодов Ю.М. Методи оптимізації в прикладах і задачах : навчальний посібник / Ю.М. Нефьодов, Т.Ю. Балицька. – Київ : Кондор, 2011. – 324 с.
2. Оптимізація систем керування.: навч. посіб. для студ. спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» / Л. Р. Ладієва; КПІ ім. Ігоря Сікорського. 2020, 211с.
3. Сікора Я. Б. Методи оптимізації та дослідження операцій [Текст] : навчальний посібник / Я. Б. Сікора, А.Й. Щехорський, Б.Л. Якимчук. – Житомир: Вид-во ЖДУ ім. Івана Франка, 2019. – 148 с.
4. Киселева Е.М. Непрерывные задачи оптимального разбиения множеств и г-алгоритмы: Монография / Е.М. Киселева, Л.С. Коряшкина. – К.: Наук. думка, 2015. – 400 с.
5. Штовба С.Д. Ш92 Методи оптимізації в середовищі MatLab. Лабораторний практикум. Навчальний посібник. – Вінниця: ВДТУ, 2001. – 56 с.

Контрольні питання

1. Сформулюйте постановку задачі безумовної оптимізації.
2. Чим методи нульового порядку відрізняються від методів першого та другого порядку?
3. Що являє собою градієнт та антиградієнт функції? Опишіть їхні властивості.
4. Чому квадратична функція використовується як основа для побудови алгоритмів нелінійної оптимізації?
5. Сформулюйте основну ідею методу Пауелла.
6. Запишіть ітераційні формули методу Пауелла.
7. Відповідно до якого принципу задається напрямок кроку у методі Пауелла?
8. Яким чином обчислюється величина кроку у методі Пауелла?
9. Який критерій закінчення обчислень у методі Пауелла?
10. Який порядок методу Пауелла? Вкажіть переваги та недоліки цього методу.
11. Дайте визначення методів сполучених градієнтів.

12. Запишіть формулу Флетчера – Рівса.
13. Опишіть метод Флетчера – Рівса.
14. Як обчислюється величина кроку у методі Флетчера – Рівса?
15. Який критерій закінчення обчислень у методі Флетчера – Рівса? Який порядок цього методу?
16. Обґрунтуйте необхідність застосування рестартів у методі Флетчера – Рівса.
17. Запишіть та обґрунтуйте формулу Полака – Ріб'єра.
18. Як обчислюється величина кроку у методі Полака – Ріб'єра? Як закінчуються обчислення у методі Полака – Ріб'єра?
19. Чим відрізняється алгоритм методу Полака – Ріб'єра від алгоритму методу Флетчера – Рівса?

Завдання для самостійної роботи

1. Розв'язати задачу оптимізації цільової функції з додатку А (за своїм варіантом) такими методами:

- 1) пошук за симплексом (метод Нелдера – Міда);
- 2) найшвидшого спуску;
- 3) Гаусса – Ньютона;
- 4) Левенберга – Марквардта;
- 5) Девідона – Флетчера – Пауела;
- 6) Бройдена – Флетчера – Гольдарба – Шенно.

2. Провести порівняльний аналіз ефективності різних методів оптимізації. Експерименти проводити для $n = 5$. Рекомендовано використовувати або бібліотеку Optimization в програмному середовищі MatLab, або SciPy в Python.

За результатами експеременту оформити звіт у такому форматі: метод; координати оптимуму; значення цільової функції в оптимумі; кількість ітерацій; час оптимізації. Зробити загальний висновок про порівняння методів.

Практичне заняття № 13

Задачі умовної оптимізації. Метод проєкції градієнта. Метод умовного градієнта

Мета: ознайомитися з методами багатовимірної умовної оптимізації: методом проєкції градієнта і методом умовного градієнта.

План заняття

1. Постановка задачі багатовимірної умовної оптимізації. Класифікація методів умовної оптимізації функцій багатьох змінних
2. Метод проєкції градієнта
3. Метод умовного градієнта

Опорні поняття

1. Постановка задачі багатовимірної умовної оптимізації. Класифікація методів умовної оптимізації функцій багатьох змінних

Загальна постановка задачі багатовимірної умовної оптимізації:

$$f(x) \rightarrow \min, x \in \Omega, \quad (13.1)$$

$$\Omega = \{x \in D: g_i(x) \leq 0, \quad 1 \leq i \leq m; \quad g_i(x) = 0, \quad m + 1 \leq i \leq s\},$$
$$D = \{x \in \mathbb{R}^n: a_j \leq x_j \leq b_j, \quad 1 \leq j \leq n\}.$$

Наближені методи умовної оптимізації функції багатьох змінних поділяють на дві групи:

- методи, що використовують перетворення задачі умовної оптимізації у послідовність задач безумовної оптимізації шляхом уведення до розгляду допоміжних функцій;
- методи безпосереднього розв'язування задачі умовної оптимізації, які ґрунтуються на русі з однієї допустимої точки, де виконані всі обмеження, до іншої із кращим значенням цільової функції.

До першої групи належать такі методи: штрафних функцій, бар'єрних функцій, множників Лагранжа; до другої групи – метод проєкції градієнта, умовного градієнта, можливих напрямків Зойтендейка.

2. Метод проєкції градієнта

Це узагальнення градієнтного методу безумовної оптимізації на випадок задач умовної оптимізації.

Розглянемо задачу

$$f(x) \rightarrow \min, x \in \Omega, \quad (13.2)$$

де Ω – замкнена, опукла множина в E^n , $f(x)$ – диференційовна функція на Ω .

Нагадаємо, що проєкцією точки a на множину Ω називається точка $P_\Omega(a)$, яка є найближчою до a серед усіх точок множини Ω . Тобто, $P_\Omega(a)$ є розв'язком задачі проєктування

$$\varphi(x) = \|x - a\| \rightarrow \min, x \in \Omega. \quad (13.3)$$

Лема 13.1. Нехай Ω – замкнена, опукла множина в E^n . Тоді

- 1) проєкція $P_\Omega(a)$ будь-якої точки $a \in \Omega$ існує і єдина;
- 2) точка \check{x} є проєкцією точки a на множині Ω ($\check{x} = P_\Omega(a)$) тоді і тільки тоді, якщо $(\check{x} - a, x - \check{x}) \geq 0$ для всіх $x \in \Omega$;
- 3) для будь-яких точок $a_1, a_2 \in E^n$ справедлива нерівність

$$\|P_\Omega(a_1) - P_\Omega(a_2)\| \leq \|a_1 - a_2\|,$$

тобто оператор проєктування має властивість нерозтягнення відстаней.

В методі проєкції градієнта за наступну точку наближення до розв'язку задачі (13.1), (13.2) вибирається

$$x^{(k+1)} = P_\Omega(x^{(k)} - \lambda_k f'(x^{(k)})), \quad k = 0, 1, 2, \dots, \quad (13.4)$$

Коефіцієнти $\lambda_k \geq 0$ можна вибирати за тими ж правилами, які сформульовані на попередніх заняттях.

Умови збіжності варіанта методу з апріорним вибором λ_k визначає:

Теорема 13.1. Нехай множина Ω є замкненою і опуклою множиною в E^n . Функція $f(x)$ – сильно опукла з константою $\theta > 0$ і диференційовна на Ω , її градієнт задовольняє умові Ліпшиця

$$\|f'(x') - f'(x'')\| \leq L\|x' - x''\|,$$

для будь-яких $x', x'' \in \Omega$.

Тоді послідовність $\{x^{(k)}\}$, яка генерується за правилом (13.4), де $x^{(0)}$ – довільна точка з Ω , а $\lambda_k = \lambda \in \left(0, \frac{4M}{L^2}\right)$, прямує до розв'язку x^* задачі (13.2), зі швидкістю геометричної прогресії, тобто

$$\|x^{(k+1)} - x^*\| \leq q \|x^{(k)} - x^*\|,$$

де $q = \sqrt{1 - 4M\lambda + \lambda^2 L^2} \in (0, 1)$.

В описаному методі на кожній k -й ітерації потрібно проводити операцію проєктування точки на множину Ω , тобто розв'язувати задачу вигляду (13.3), коли $a = x^{(k)} - \lambda_k f'(x^{(k)})$. У деяких випадках вдається вказати явний вигляд проєкції, приміром:

- 1) якщо $\Omega = \{x \in E^n: \|x - c\| \leq R\}$ – куля, то

$$P_\Omega(a) = c + R \frac{a - c}{\|a - c\|};$$

2) якщо $\Omega = \{x \in E^n: b_j \leq x_j \leq c_j, j = \overline{1, n}\}$ – координатний паралелепіпед, то

$$P_{\Omega}(a) = w = (w_1, w_2, \dots, w_n);$$

$$w_j = \begin{cases} b_j, & a_j < b_j, \\ a_j, & b_j < a_j < c_j, \\ c_j, & a_j > c_j; \end{cases}$$

3) якщо $\Omega = \{x \in E^n: x_j \geq 0, j = \overline{1, n}\}$ – невід’ємний ортант, то

$$P_{\Omega}(a) = (\max(0, a_1), \max(0, a_2), \dots, \max(0, a_n));$$

4) якщо $\Omega = \{x \in E^n: (p, x) \geq \beta, p \neq 0\}$ – півпростір, то

$$P_{\Omega}(a) = a + \max(0, \beta - (p, a)) \frac{p}{\|p\|^2};$$

1) якщо $\Omega = \{x \in E^n: Ax = b\}$ – афінна множина, при цьому рядки матриці A лінійно незалежні, то

$$P_{\Omega}(a) = a - A^T(A \cdot A^T)^{-1}(Aa - b),$$

де A^T – транспонована матриця.

Але, якщо множина Ω задається за допомогою більш-менш складної системи рівностей та нерівностей, то метод проєкції градієнта застосувати практично неможливо, оскільки задача (13.3) має таку ж складність, як і початкова.

Використовуючи ідею проєктування, можна модифікувати інші методи безумовної оптимізації, застосовуючи їх до задач умовної оптимізації, у тому числі метод Ньютонa, методи спряжених напрямків та інші.

Метод проєкції градієнта Розена

Тут антиградієнт проєктується таким чином, що значення цільової функції покращується і в той же час зберігається допустимість точок траєкторії.

Припустимо, що всі обмеження $g_i(x) \geq 0, i = \overline{1, n}$ – лінійні, та перепишемо їх у вигляді

$$a_i x \leq b_i, i \in I_1, \quad (13.5)$$

$$a_i x = b_i, i \in I_2, \quad (13.6)$$

де $a_i = \{a_{i1}, a_{i2}, \dots, a_{in}\}$ – вектори-стовпці, I_1, I_2 – множини індексів.

Нехай допустимий розв’язок $x \in$ відомим, а $I_0(x) = \{i: a_i x = b_i, i \in I_1\} \cup I_2$ – це множина індексів обмежень, які є активними в точці x . Знайдемо напрямок p ($\|p\| = 1$), який дозволяє зменшити значення цільової функції $f(x)$

(мінімізувати $f'(x) \cdot p$) й залишитися в множині допустимих розв'язків. Це означає, що напрямок p повинен задовольняти таким співвідношенням:

$$a_i p = 0, \forall i \in I_0(x). \quad (13.7)$$

Позначимо через A_0 підматрицю матриці обмежень A , яка складається з рядків $i \in I_0(x)$ матриці A . Припустимо, що ранг матриці A_0 становить q і дорівнює кількості елементів множини $I_0(x)$.

Теорема 13.2. Якщо A_0 є матриця вимірності $q \times n$ і рангу $q \leq n$, то оптимальним розв'язком задачі

$$\begin{aligned} f'(x) \cdot p &\rightarrow \min, \\ A_0 p &= 0, \|p\| = 1 \end{aligned}$$

є вектор $p = \frac{y}{\|y\|}$, де y – проєкція вектора $(-f'(x))$ на $S_0 = \{y : A_0 y = 0\}$, яка визначається за такою формулою:

$$y = -P_0 f'(x) = -(I - A_0^T [A_0 A_0^T]^{-1} A_0) f'(x).$$

P_0 називається матрицею проєктування.

Зауважимо, що, коли вектор $y \neq 0$, він є можливим напрямком спуску, оскільки

$$f'(x)y = -y^T y < 0.$$

Якщо $y = -P_0 f'(x) = 0$ у поточній точці x , то

$$f'(x) + A_0^T u = 0, \quad (13.8)$$

$$u = -[A_0 A_0^T]^{-1} A_0 f'(x). \quad (13.9)$$

Оскільки стовпці матриці A_0^T є градієнтами активних обмежень у точці x , то ясно, що коли $u \geq 0$ співвідношення (13.8) відповідає умовам Куна – Такера у поточній точці x , а це – рівносильно твердженню, що точка x є точкою локального оптимуму задачі.

У випадку, коли $y = 0$, але вектор u , який визначається співвідношенням (13.9), має строго від'ємні компоненти, поточна точка може не бути оптимальною і необхідно шукати інший напрямок переміщення.

Для цього виключимо з $I_0(x)$ одне з обмежень, для яких $u_i < 0$ (наприклад, те, у якому u_i має найбільше значення за модулем). При цьому отримаємо нову матрицю проєктування P_0 , яка дозволить знайти новий напрямок переміщення $y = -P_0 f'(x)$.

Позначимо через $\bar{I}_0(x)$ множини обмежень неактивних у допустимій точці x :

$$\bar{I}_0(x) = \{i : a_i x = b_i, i \in I_1\}, A_1 = \{a_i\}, B_1 = \{b_i\}, i \in \bar{I}_0(x).$$

Для кращого розуміння методу проєкції градієнта Розена розглянемо його геометричну ілюстрацію (рис. 13.1). Припустимо, що отримано допустимий

розв'язок $x^{(k)}$. Ми не можемо рухатися в напрямку градієнта, не порушуючи обмеження 1. Матриця A_0 , яка використовується в методі, дорівнює a_1 . Як показано на рис. 13.1, вектор $p^{(k)}$ є ортогональною проекцією вектора $(-f'(x^{(k)}))$ на границю множини допустимих розв'язків. Відтак, на наступному кроці потрібно рухатися вздовж границі, доки не буде досягнута крайня точка A , в якій ітераційний процес закінчиться.

Розглянемо тепер ситуацію, яку зображено на рис. 13.2. У цьому прикладі будуть порушуватись обидва обмеження, якщо рухатися в напрямку градієнта. Ми можемо знайти напрямок руху, якщо припустимо, що обмеження $a_2 p \leq 0$ виконувалося би як нерівність. Аби отримати матрицю A_0 , викреслимо рядок, який відповідає a_2 з A_0 , і прийдемо до попереднього випадку.

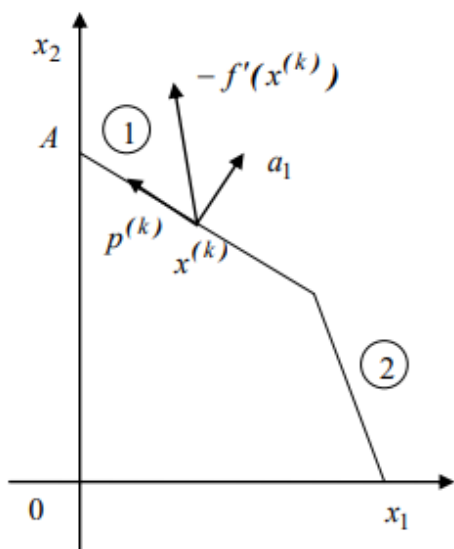


Рис. 13.1.

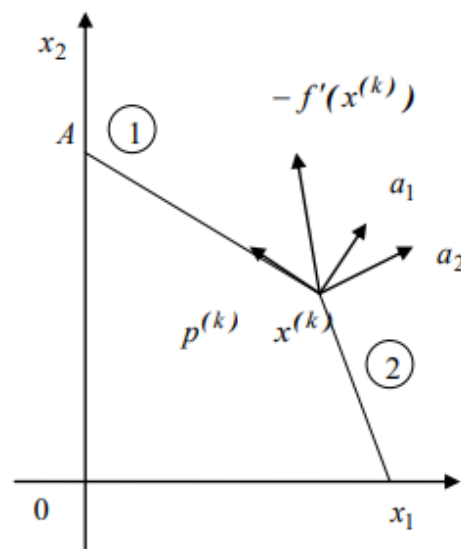


Рис. 13.2

Алгоритм методу проєкції градієнта Розена

Початковий етап. Задати допустиму точку $x^{(0)}$, вважати, що $k = 0$, та перейти до основного етапу.

Основний етап

Крок 1. Визначити множину $I_0(x^{(k)})$ індексів активних обмежень та множину індексів неактивних обмежень $\bar{I}_0(x^{(k)})$.

Крок 2. Нехай A_0 – матриця, рядки якої відповідають обмеженням $i \in I_0(x^{(k)})$, а A_1 – матриця, рядки якої відповідають обмеженням $i \in \bar{I}_0(x^{(k)})$. Якщо A_0 порожня, то вважати, що $P_0 = I$. У протилежному випадку обчислити матрицю проєктування

$$P_0 = I - A_0^T [A_0 A_0^T]^{-1} A_0 \text{ та вектор } p^{(k)} = -P_0 f'(x^{(k)}).$$

Якщо $p^{(k)} = 0$, то перейти на крок 4.

Крок 3. Якщо $p^{(k)} \neq 0$, то знайти

$$\alpha_{max} = \begin{cases} \min_i \left(\frac{e_i}{d_i}, d_i > 0 \right), & d > 0, \\ \infty, & d \leq 0 \end{cases} \quad (13.10)$$

де $e = B_1 - A_1 x^{(k)}$, $d = A_1 p^{(k)}$.

Знайти таке $x^{(k+1)}$, при якому $f(x^{(k+1)}) = \min_{0 \leq \alpha \leq \alpha_{max}} f(x^{(k)} + \alpha p^{(k)})$.

Вважати, що $x^{(k+1)} = x^{(k)} + \alpha_k p^{(k)}$, $k = k + 1$, та перейти до кроку 1.

Крок 4. Якщо $p^{(k)} = 0$ і матриця A_0 порожня, то зупинитися.

Нехай

$$u = -[A_0 A_0^T]^{-1} A_0 f'(x^{(k)}).$$

Якщо $u \geq 0$, то зупинитися: $x^{(k)}$ задовольняє умовам Куна – Такера. У протилежному випадку вважати, що $I_0(x^{(k)}) = I_0(x^{(k)}) - \{i\}$, де індекс i відповідає найбільшій за модулем з від’ємних компонент u_i вектора u . Перейти до кроку 1.

Алгоритм описаний.

Реалізація методу Розена на ЕОМ передбачає можливість виникнення додаткових труднощів. Наприклад, навіть, якщо $p^{(k)} = 0$, то при наявності помилок округлення вектор $p^{(k)}$, який обчислений на комп’ютері, може відрізнитися від нульового. Отже, потрібно використовувати інше правило переходу до викреслювання рядку з A_0 . Одне з правил таке: завжди потрібно обчислювати

$$f'(x^{(k)}) \cdot p^{(k)}, p^{(k)} = -P_0 f'(x^{(k)}) \quad \text{та} \quad f'(x^{(k)}) \cdot \bar{p}^{(k)}, \bar{p}^{(k)} = -\bar{P}_0 f'(x^{(k)}),$$

де \bar{A}_0 може бути отримана з A_0 шляхом викреслювання рядка, який відповідає найбільшому за модулем від’ємному u_i , коли є хоча б одне від’ємне u_i .

Потім потрібно вибрати напрямок, який відповідає більшому з цих двох чисел.

3. Метод умовного градієнта

Його ще називають методом лінійної апроксимації (лінеаризації) цільової функції.

Розглянемо задачу:

$$f(x) \rightarrow \min, x \in \Omega \subset E^n, \quad (13.11)$$

де Ω – опукла, замкнена, обмежена множина в E^n , $f(x) \in C^1(\Omega)$.

Ітераційна формула методу умовного градієнта має такий вигляд:

$$x^{(k+1)} = x^{(k)} + \alpha_k h^{(k)}, k = 0, 1, 2, \dots, \quad (13.12)$$

де $h^{(k)}$ – напрямок спуску цільової функції $f(x)$ у точці $x^{(k)}$, α_k – параметр, який регулює довжину кроку вздовж $h^{(k)}$.

Для вибору $h^{(k)}$ на k -тій ітерації розв'язують задачу мінімізації на множині Ω лінійної апроксимації цільової функції $f(x)$ в точці $x^{(k)}$, тобто такої функції:

$$f_k(x) = f(x^{(k)}) + (f'(x^{(k)}), x - x^{(k)}),$$

де $(f'(x^{(k)}), x - x^{(k)})$ – головна лінійна частина приросту цільової функції $f(x)$ у точці $x^{(k)}$. Відкидаючи константу $f(x^{(k)})$, цю задачу можна записати у такому вигляді:

$$(f'(x^{(k)}), x - x^{(k)}) \rightarrow \min, x \in \Omega. \quad (13.13)$$

Нехай $\bar{x}^{(k)}$ – оптимальний розв'язок задачі (13.13), а $f_k(\bar{x}^{(k)}) = (f'(x^{(k)}), \bar{x}^{(k)} - x^{(k)})$ – оптимальне значення її цільової функції. Згідно з теоремою Веєрштраса розв'язок $\bar{x}^{(k)}$ задачі (13.12) завжди існує. Якщо задача (13.13) має декілька розв'язків, то обирається один з них. Враховуючи, що $x^{(k)} \in \Omega$, маємо

$$\min_{x \in \Omega} f_k(x) = f_k(\bar{x}^{(k)}) \leq f_k(x^{(k)}) = 0.$$

Тому можливі лише два випадки: $f_k(\bar{x}^{(k)}) = 0$ або $f_k(\bar{x}^{(k)}) < 0$.

Якщо $f_k(\bar{x}^{(k)}) = 0$, то $(f'(x^{(k)}), x - x^{(k)}) \geq f_k(\bar{x}^{(k)}) = 0$ для будь-яких $x \in \Omega$, тобто точка $x^{(k)}$ є стаціонарною для задачі (13.11) відповідно до теореми про необхідні умови оптимальності в задачі умовної мінімізації. Робота алгоритму завершується, точку $x^{(k)}$ необхідно дослідити на оптимальність.

Якщо функція $f(x)$ є опуклою на множині Ω , то згідно з критерієм оптимальності першого порядку точка $x^{(k)}$ – розв'язок задачі (13.11).

Нехай тепер $f_k(\bar{x}^{(k)}) < 0$. У цьому випадку $\bar{x}^{(k)} \neq x^{(k)}$. Тоді у формулі (13.12) будемо вважати, що $h^{(k)} = \bar{x}^{(k)} - x^{(k)}$, і ітераційна формула (13.12) набуде такого вигляду:

$$x^{(k+1)} = x^{(k)} + \alpha_k(\bar{x}^{(k)} - x^{(k)}) \quad k = 0, 1, 2, \dots \quad (13.14)$$

Вектор $h^{(k)}$ прийнято називати *умовним антиградієнтом* цільової функції $f(x)$ у точці $x^{(k)}$. Оскільки множина допустимих розв'язків Ω є опуклою, то для будь-якого α_k з відрізка $[0, 1]$ точка $x^{(k+1)} = x^{(k)} + \alpha_k(\bar{x}^{(k)} - x^{(k)}) \in \Omega$.

Опишемо способи вибору крокового множника α_k .

1. Визначаємо α_k з умови:

$$g_k(\alpha_k) = \min_{0 \leq \alpha \leq 1} g_k(\alpha), \quad g_k(\alpha) = f(x^{(k)} + \alpha(\bar{x}^{(k)} - x^{(k)})). \quad (13.15)$$

2. Параметр α_k вибираємо апріорно, тобто

$$0 < \alpha_k \leq 1, \quad k = 0, 1, 2, \dots, \quad \sum_{k=0}^{\infty} \alpha_k = \infty, \quad \sum_{k=0}^{\infty} (\alpha_k)^2 < \infty.$$

Наприклад, за α_k можна узяти $\alpha_k = \frac{1}{k+1}$.

3. Адаптивний (автоматичний) вибір крокового множника. Параметр α_k вибирається за правилом дроблення, допоки не виконається нерівність:

$$f\left(x^{(k)} + \alpha(\bar{x}^{(k)} - x^{(k)})\right) - f(x^{(k)}) \leq \delta \alpha f_k(\bar{x}^{(k)}),$$

де δ – параметр методу, $0 < \delta < 1$.

4. Дроблення кроку. Вибираємо $\alpha_k = 1$ і перевіряємо виконання умови монотонності: $f(x^{(k+1)}) < f(x^{(k)})$. Потім при необхідності дробимо кроковий множник α_k ($\alpha_k = \lambda \alpha_k$, $0 < \lambda < 1$), допоки не буде виконана умова монотонності.

Умови збіжності методу умовного градієнта

Теорема 13.3. Нехай Ω – замкнена, обмежена, опукла множина в E^n , $f(x) \in C^1(\Omega)$, причому її градієнт задовольняє умові Лібшиця, тобто

$$\|f'(x) - f'(y)\| \leq L\|x - y\| \quad \forall x, y \in \Omega.$$

Тоді будь-яка гранична точка x^* послідовності $\{x^{(k)}\}_{k=0}^{\infty}$, яка визначається умовами (13.13) – (13.15), є стаціонарною для задачі (13.11), тобто

$$(f'(x^*), x - x^*) \geq 0 \quad \text{для } \forall x \in \Omega.$$

Якщо при цьому цільова функція $f(x)$ опукла на множині Ω , то x^* – розв'язок задачі (13.11) і

$$\lim_{k \rightarrow \infty} f(x^{(k)}) = f_*, \quad \text{де } f_* = \min_{x \in \Omega} f(x).$$

Зауваження 1. Аналогічні теореми можна сформулювати і для інших способів вибору α_k . Критерії завершення ітераційного процесу:

$$|f_k(\bar{x}^{(k)})| < \varepsilon;$$

$$\|x^{(k+1)} - x^{(k)}\| < \varepsilon.$$

Опишемо алгоритм методу умовного градієнта.

Початковий етап. Вибрати початкову точку $x^{(0)} \in \Omega$. Задати константу δ ($0 < \delta < 1$), константу зупинки $\varepsilon > 0$, передбачити $k = 0$ та перейти до основного етапу.

Основний етап

Крок 1. Обчислити $f'(x^{(k)})$. Знайти оптимальний розв'язок $\bar{x}^{(k)}$ задачі

$$(f'(x^{(k)}), x - x^{(k)}) \rightarrow \min, \quad x \in \Omega.$$

Крок 2. Якщо $f_k(\bar{x}^{(k)}) = (f'(x^{(k)}), \bar{x}^{(k)} - x^{(k)}) = 0$, то зупинитися: $x^{(k)}$ – стаціонарна точка, інакше обчислити умовний антиградієнт $h^{(k)} = \bar{x}^{(k)} - x^{(k)}$, вважати, що $\alpha_k = 1$.

Крок 3. Перевірити виконання умови

$$f(x^{(k)} + \alpha(\bar{x}^{(k)} - x^{(k)})) - f(x^{(k)}) \leq \delta \alpha f_k(\bar{x}^{(k)}). \quad (13.16)$$

Якщо нерівність виконується, то перейти до кроку 4, інакше провести дроблення крокового множника α_k , доки не буде виконано нерівність (13.16).

Крок 4. Обчислити наступне наближення $x^{(k+1)} = x^{(k)} + \alpha_k h^{(k)}$, замінити k на $k + 1$. Якщо $\|x^{(k+1)} - x^{(k)}\| < \varepsilon$, то зупинитися, у протилежному випадку перейти до кроку 1.

Алгоритм описаний.

Зауваження 2. Для деяких множин Ω розв'язок задачі (13.14) можна отримати в аналітичному вигляді, наприклад, коли Ω – куля або паралелепіпед у n -вимірному просторі.

Завдання. Запишіть аналітичний розв'язок задачі (13.14) у випадку, коли Ω – куля або паралелепіпед у n -вимірному просторі.

Задачі для розв'язування у класі

Визначити, який метод можна застосовувати до розв'язання поданої нижче задачі умовної оптимізації і виконати 2 – 3 ітерації цього методу.

1. $f = 4(x_1 + 1)^2 + (x_2 - 14)^2 \rightarrow \text{extr}$
 $x_1 + x_2 - 10 = 0$

2. $f = (x_1 + 2)^2 + 4(x_2 - 1)^2 \rightarrow \text{extr}$
 $(x_1 - 4)^2 + (x_2 + 1)^2 \leq 9$

3. $f = 4/x_1 + 9/x_2 + x_1 + x_2 \rightarrow \text{min}$
 $4x_1 + 5x_2 \leq 20$
 $x_j \geq 0, j = \overline{1,2}$

4. $z = -x_1^2 - 2x_2^2 + 2x_1 + 12x_2 \rightarrow \text{max}$
 $4x_1 + 5x_2 \leq 40$
 $x_1 - 5x_2 = 10,$
 $x_j \geq 0, j = \overline{1,2}$

Рекомендована література та інформаційні ресурси

1. Нефьодов Ю.М. Методи оптимізації в прикладах і задачах : навчальний посібник / Ю.М. Нефьодов, Т.Ю. Балицька. – Київ : Кондор, 2011. – 324 с.
2. Оптимізація систем керування.: навч. посіб. для студ. спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» / Л. Р. Ладієва; КПІ ім. Ігоря Сікорського. 2020, 211с.
3. Жалдак М.І. Основи теорії і методів оптимізації: Навчальний посібник / М.І. Жалдак, Ю.В. Триус. – Черкаси: Брама-Україна, 2005. – 608 с.

4. Сікора Я. Б. Методи оптимізації та дослідження операцій [Текст] : навчальний посібник / Я. Б. Сікора, А.Й. Щехорський, Б.Л. Якимчук. – Житомир: Вид-во ЖДУ ім. Івана Франка, 2019. – 148 с.

5. Кононюк А.Е. Основы теории оптимизации. Безусловная оптимизация К.2.ч.1. / А.Е. Кононюк. – Киев:"Освіта України", 2011. – 544 с.

6. Кіселева О.М. Неперервні задачі оптимального розбиття множин та ϵ -алгоритми: Монографія / О.М. Кіселева, Л.С. Коряшкіна.– К.: Наук. думка, 2015. – 400 с.

Контрольні питання

1. За яких умов задачу оптимізації можна розв'язувати методом проєкції градієнта?

2. Сформулюйте основну ідею методу проєкції градієнта.

3. За яких умов задачу оптимізації можна розв'язувати методом умовного градієнта?

4. Яка основна ідея методу умовного градієнта?

5. Назвіть критерії закінчення ітераційного процесу, які можна використовувати в методі проєкції градієнта?

6. Опишіть критерії закінчення ітераційного процесу, котрі можна використовувати в методі умовного градієнта? Який з них властивий лише для цього методу?

7. Яким чином можна задавати або змінювати параметр довжини кроку в методі проєкції градієнта?

8. Як задавати або змінювати параметр довжини кроку в методі умовного градієнта?

Індивідуальне завдання

Реалізувати метод проєкції градієнта і метод умовного градієнта для розв'язання задачі умовної оптимізації

$$f = (x_1 - N)^2 + (N + 25)(x_2 + N - 5)^2 \rightarrow \min$$

$$(x_1 - 4)^2 + (x_2 + 3)^2 \leq 16,$$

де N – номер студента в журналі академічної групи.

Практичне заняття № 14

Задачі умовної оптимізації. Виключення обмежень за допомогою заміни змінних. Методи штрафних функцій

Мета: ознайомитися з методами виключення обмежень, навчитися використовувати методи штрафних функцій та модифікованих функцій Лагранжа для розв'язання задач умовної оптимізації.

План заняття

1. Постановка задачі умовної оптимізації.
2. Виключення обмежень за допомогою заміни змінних.
3. Ідея методу штрафних функцій. Метод зовнішнього штрафу та бар'єрних функцій.
4. Метод модифікованих функцій Лагранжа.

Опорні поняття

1. Постановка задачі умовної оптимізації

Загальна постановка задачі нелінійного програмування:

$$f(x) \rightarrow \min, x \in \Omega, \quad (14.1)$$

$$\Omega = \{x \in D: g_i(x) \leq 0, \quad 1 \leq i \leq m; \quad g_i(x) = 0, \quad m+1 \leq i \leq s\},$$
$$D = \{x \in \mathbb{R}^n: a_j \leq x_j \leq b_j, \quad 1 \leq j \leq n\}.$$

2. Виключення обмежень за допомогою заміни змінних

Формально найпростіше виключити прямі обмеження. Для цього достатньо виконати заміну змінних за однією з формул табл.14.1, де z_j суть нові незалежні змінні. Однак, подібна заміна змінних в деяких випадках призводить до значної обчислювальної складності. Зокрема, може зростати обумовленість матриць Гессе цільової функції за новими змінними, і, навіть, можуть бути отримані вироджені матриці.

Задачі параметричної ідентифікації, що виникають під час алгоритмізації процесів керування, найбільш часто містять складні нелінійні умови. Це не дозволяє використовувати спеціальну техніку оптимізації, спрямовану, наприклад, на розв'язання задач з лінійними або прямими обмеженнями.

Заміна змінних для виключення обмежень

№	Обмеження	Перетворення
1	$x_j > a_j$	$x_j = a_j + \exp(z_j)$
2	$x_j > a_j$	$x_j = a_j + z_j^2$
3	$x_i > x_j, i \neq j$	$x_j = z_j, x_i = z_j + z_i^2$
4	$a_j \leq x_j \leq b_j$	$x_j = b_j + (a_j - b_j)\sin^2(z_j)$
5	$a_j \leq x_j \leq b_j$	$x_j = 0,5(a_j + b_j) + 0,5(b_j - a_j)\sin(z_j)$
6	$a_j < x_j < b_j$	$x_j = b_j + (a_j - b_j)\frac{1}{\pi}\operatorname{arcctg}(z_j)$
7	$a_j < x_j < b_j$	$x_j = b_j + (a_j - b_j)\exp(z_j)/(1 + \exp(z_j))$
8	$a \leq x_j, x_i \leq b$ $x_i > x_j, i \neq j$	$x_j = b + (a - b)\sin^2(z_j)$ $x_i = b + (a - b)\sin^2(z_j)\sin^2(z_i)$
9	$a < x_j < b$ $a < x_i < b$ $x_i > x_j, i \neq j$	$x_j = b + (a - b)\frac{1}{\pi}\operatorname{arcctg}(z_j)$ $x_i = b + (a - b)\frac{1}{\pi^2}\operatorname{arcctg}(z_j)\operatorname{arcctg}(z_i)$
10	$a_i(x_j) \leq x_i \leq b_i(x_j)$ $i \neq j$	$x_j = z_j$ $x_i = b_i(z_j) + (a_i(z_j) - b_i(z_j))\sin^2(z_i)$
11	$a_i(x_k) \leq x_i \leq b_i(x_j)$ $i \neq j, i \neq k$	$x_j = z_j$ $x_k = z_k$ $x_i = b_i(z_j) + (a_i(z_k) - b_i(z_j))\sin^2(z_i)$

Існує два класи оптимізації, орієнтовані на розв'язання задач з обмеженнями загального вигляду. Перший містить алгоритми, які реалізують методи проєкції градієнта, відсікання, а також різні методи можливих напрямків. Вони дають можливість на кожній ітерації звести вихідну задачу до формально більш простої, але з обмеженнями, наприклад, до задачі лінійного програмування. Другу групу формують методи штрафних функцій і модифікованих функцій Лагранжа, засновані на врахуванні обмежень безпосередньо в конструкції критерію оптимальності з подальшим використанням алгоритмів безумовної оптимізації. Далі будемо розглядати другий, у певному сенсі, більш перспективний підхід.

3. Ідея методу штрафних функцій. Метод зовнішнього штрафу та бар'єрних функцій

Метод штрафних функцій є одним з найпростіших і найпоширеніших для розв'язування задачі математичного програмування. Основна його ідея полягає в тому, що задача мінімізації певної функції за наявності обмежень зводиться до еквівалентної задачі мінімізації деякої допоміжної функції без обмежень або до послідовності таких завдань. При цьому допоміжна функція підбирається так, аби вона збігалася із заданою функцією всередині допустимої області і швидко зростала поза нею.

Проілюструємо ідею методу штрафних функцій.

Приклад 14.1. Мінімізувати функцію

$$f(x) = (x_1 - 4)^2 + (x_2 - 5)^2$$

за умов

$$g(x) = x_1 + x_2 - 6 = 0.$$

Розв'язування. Додаємо $g^2(x)$ до цільової функції, внаслідок чого отримуємо нову функцію, а саме:

$$P(x) = (x_1 - 4)^2 + (x_2 - 5)^2 + (x_1 + x_2 - 6)^2.$$

Значення цієї функції вважатимемо вільними від будь-яких обмежень, нова складова тут виступає в ролі штрафу. В процесі мінімізації функції $P(x)$ ця штрафна добавка до $f(x)$ сприяє більш точному виконанню для вектора x вихідного обмеження $g(x) = 0$. Цілком очевидно, що коли умови $g(x) = 0$ виконуються із заданим ступенем точності, то величина штрафного доданка є зневажливо малою і, отже, значення допоміжної функції буде близьким до значення вихідної функції $f(x)$. Іншими словами, вплив обмеження, котре включене до функції $P(x)$, з розвитком процесу оптимізаційного пошуку весь час слабшає і на нескінченності повністю зникає. Тому послідовність проміжних значень функції $P(x)$ збігається до того ж значення, що й відповідна послідовність значень функції $f(x)$, отже, екстремум функції $P(x)$ співпадає з екстремумом функції $f(x)$.

Перевага, яку ми отримуємо за рахунок переходу від задачі мінімізації за наявності обмежень до задачі без обмежень, полягає в тому, що в останньому випадку мінімізація може здійснюватися за допомогою значно простіших, у порівнянні з першим випадком, алгоритмів. На рис. 14.1 показана задача мінімізації функції $f(x)$ за наявності обмеження $g(x) = 0$, на рис. 14.2 – мінімізація допоміжної цільової функції $P(x)$.

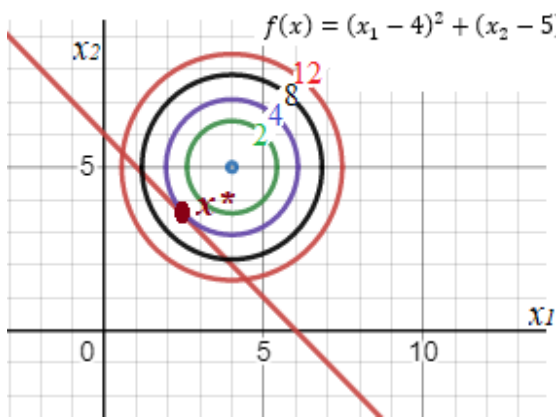


Рис. 1 Задача умовної оптимізації

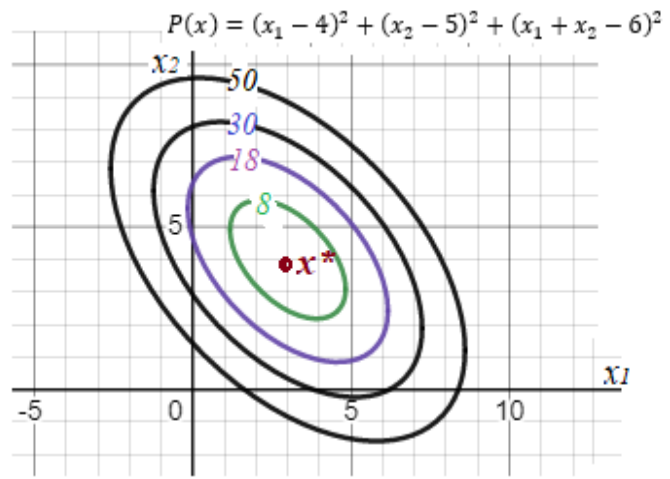


Рис. 2 Лінії рівня допоміжної функції

Всі методи штрафних функцій можна розділити на два класи: параметричні і непараметричні.

Параметричні методи характеризуються наявністю одного або кількох належним чином підібраних параметрів, що входять до структури штрафної функції (яка будується за допомогою обмежень) у вигляді вагових коефіцієнтів.

Їх поділяють на три типи методів: 1 – внутрішньої точки; 2 – зовнішньої точки; 3 – комбіновані (внутрішньої та зовнішньої точок).

Методи внутрішньої точки характеризуються тим, що за рахунок вибору штрафної функції можна досягти того, аби усі наближення до оптимуму перебували всередині допустимої області.

Методи зовнішньої точки, навпаки, генерують послідовність точок, яка виходить за межі допустимої області, але прямує до допустимого розв'язку. Штрафна функція в цьому випадку не дозволяє наближенням надто віддалятися від границі допустимої області.

У *комбінованих методах* (використання яких доцільне, коли обмеження мають вигляд рівностей) у ході мінімізації одні з умов, котрі обмежують допустиму множину, задовольняються, а інші не задовольняються, проте, всі умови в межах заданої точності виявляються задоволеними при досягненні шуканого розв'язку.

Метод штрафних функцій в задачах нелінійного програмування з обмеженнями у формі нерівностей

Постановка задачі:

$$f(x) \rightarrow \min, x \in \Omega = \{x \in E_n, g_s(x) \leq 0, s = \overline{1, p}\}. \quad (14.2)$$

Для ілюстрації ідеї методу штрафних функцій побудуємо нову функцію

$$f_\infty(x) = f(x) + P_\infty(x),$$

де

$$P_\infty(x) = \begin{cases} 0, & x \in \Omega \\ \infty, & x \notin \Omega. \end{cases}$$

Вочевидь, x^* є розв'язком задачі (14.2) тоді і тільки тоді, коли вона мінімізує функцію $f_\infty(x)$ на всьому просторі E_n , тобто задача (14.2) еквівалентна задачі безумовної мінімізації функції $f_\infty(x)$. Перевагою останньої є те, що вона може бути розв'язана за допомогою більш простих (у порівнянні з першою задачею) алгоритмів. На рис. 14.3 зображено випадок, коли функція $f(x)$ є лінійною, а $\Omega = [a, b] \subset E_1$, тобто обмеження мають такий вигляд:

$$g_1(x) = x - b \leq 0, \quad g_2(x) = a - x \leq 0.$$

У виразі $f_\infty(x)$ можна замінити $P_\infty(x)$ на

$$P_L(x) = \begin{cases} 0, & x \in \Omega, \\ L, & x \notin \Omega, \end{cases}$$

де L – велика величина. Однак отриману функцію складно мінімізувати, оскільки вона є розривною, і велике L може викликати похибки округлень. Тому в методі штрафних функцій замість $P_\infty(x)$ розглядається послідовність функцій $P_k(x)$, яка прямує до $P_\infty(x)$, коли k прямує до нескінченності.

Доданок $P_k(x)$ слугує «штрафом» за порушення умови $x \in \Omega$: якщо $x \in \Omega$, то $P_k(x) = 0$; якщо $x \notin \Omega$, то $P_k(x) > 0$, і навіть за невеликого порушення обмеження функцію $P_k(x)$ можна зробити дуже великою за рахунок вибору параметра k . А, отже, можна очікувати, що точка мінімуму x^{k*} функції $f_k(x) = f(x) + P_k(x)$ на E_n при достатньо великих k є достатньо близькою до точки мінімуму x^* задачі (14.2).

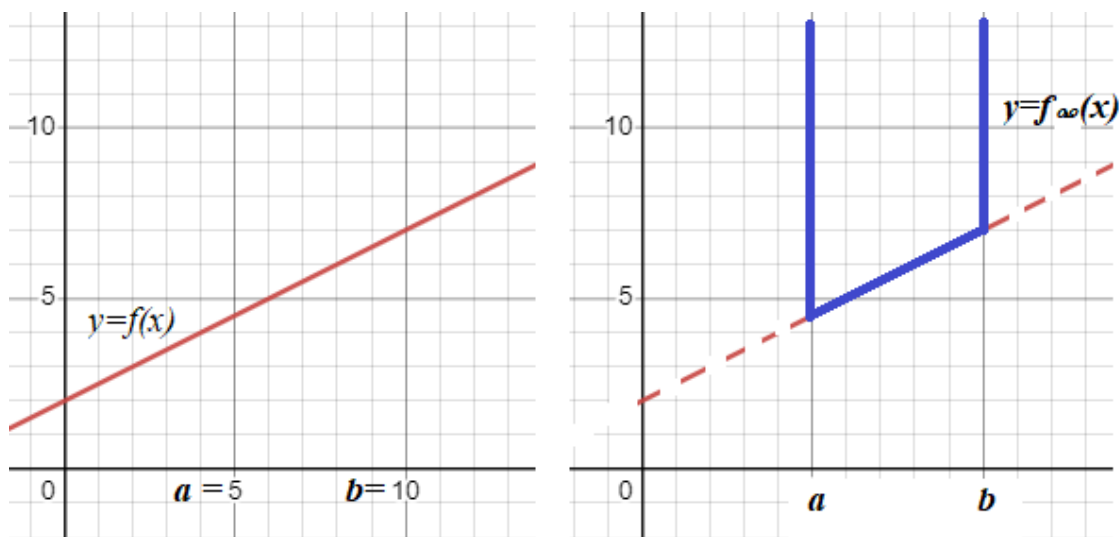


Рис. 14.3 Цільова функція в одновимірній умовній оптимізації (ліворуч) і розширена цільова функція в еквівалентній задачі безумовної оптимізації (праворуч)

Визначення 14.1. Функція $P_k(x)$ називається штрафною функцією множини Ω , якщо:

$$1) \quad P_k(x) \geq 0 \quad \forall x \in E_n \quad \text{і} \quad k > 0;$$

$$2) \lim_{k \rightarrow \infty} P_k(x) = \begin{cases} 0, & x \in \Omega, \\ \infty, & x \notin \Omega. \end{cases}$$

Відтак, метод штрафних функцій для розв'язання задачі нелінійного програмування (14.2) передбачає заміну цієї задачі послідовністю задач безумовної оптимізації функції $f_k(x) = f(x) + P_k(x)$.

Оскільки більшість методів безумовної оптимізації призначено для знаходження локальних екстремумів, то всюди у подальшому будемо говорити про пошук локального мінімуму в задачі нелінійного програмування.

Обґрунтування методу штрафних функцій надає наступна теорема.

Теорема 14.1. Припустимо цільова функція $f(x)$ задачі (14.2) є неперервною, а штрафна функція $P_k(x)$ має такі властивості:

1. $P_k(x)$ неперервна поза обома змінними x та k і $P_k(x) = 0$, коли $x \in \Omega$;
2. $P_k(x)$ монотонно зростає із зростанням k .

Нехай, крім того, для довільної константи C множина

$$\Omega_C(k) = \{x: x \in E_n, f_k(x) \leq C\}$$

є обмеженою. Тоді:

- 1) функція $f_k(x)$ досягає на E_n свого мінімуму f_k^* в деякій точці x_k^* , при цьому $f_k^* \leq f^*$, де $f^* = \min_{x \in \Omega} f(x)$ й $f_k^* \rightarrow f^*$;
- 2) будь-яка гранична точка x^* послідовності $\{x_{k_i}^*\}$ ($k_i \rightarrow \infty$ при $i \rightarrow \infty$) є точкою мінімуму $f(x)$ на Ω ;
- 3) у випадку єдиності точки мінімуму x^* функції $f(x)$ на Ω послідовність $\{x_{k_i}^*\}$ прямує до x^* .

Способи задання штрафної функції. Наведемо деякі найбільш використовувані функції штрафу для множини Ω задачі (14.2):

$$P_k(x) = k \sum_{s=1}^p [\max \{g_s(x), 0\}]^2, \quad (14.3)$$

$$P_k(x) = k \sum_{s=1}^p \max \{g_s(x), 0\}, \quad (14.4)$$

$$P_k(x) = \frac{1}{k} \exp[\max \{g_s(x), 0\}] \quad (14.5)$$

$$P_k(x) = k \max_{1 \leq s \leq p} \{\max \{g_s(x), 0\}\} \quad (14.6)$$

$$P_k(x) = \frac{1}{k} \sum_{s=1}^p \frac{1}{-g_s(x)}, \quad (14.7)$$

$$P_k(x) = \frac{1}{k} \sum_{s=1}^p \frac{1}{[g_s(x)]^2}, \quad (14.8)$$

$$P_k(x) = -\frac{1}{k} \sum_{s=1}^p \ln(-g_s(x)). \quad (14.9)$$

Функції (14.7) – (14.9) відрізняються від функцій (14.3) – (14.6) тим, що, по-перше, вони визначені лише на множині внутрішніх точок Ω ; по-друге, вони прямують до нескінченності, коли x прямує до границі Ω .

Вибір функції має бути узгоджений з методами мінімізації розширеної функції. Тобто, її варто обирати залежно від методу, який застосовується для мінімізації $f_k(x) = f(x) + P_k(x)$ на E_n . Під час вибору штрафної функції слід враховувати її гладкість, можливість просто розрахувати значення функцій і її похідних та таке інше. Наприклад, якщо задачі (14.2) відноситься до опуклого програмування, то функцію $P_k(x)$ зручно вибирати у вигляді (14.3), оскільки вона також буде опуклою (як лінійна комбінація опуклих функцій з невід'ємними коефіцієнтами), і, отже, задача мінімізації $f_k(x)$ стає задачею мінімізації опуклої функції на всьому просторі E_n .

Приклад 14. 2. Мінімізувати функцію $f(x) = x^2 - 2$, коли $x \leq 1$. Очевидно, $x^* = 1$. Скористаємося функцією штрафу вигляду (14.3) і запишемо розширену функцію, а саме:

$$f_k(x) = x^2 - 2 + k[\max\{x - 1, 0\}]^2.$$

Функція $f_k(x)$ є опуклою і неперервно диференційовною, відтак, вона досягає мінімуму в єдиній точці, яка є розв'язком рівняння

$$\frac{df_k(x)}{dx} = 0.$$

Метод зовнішньої точки

Розглянемо геометричну інтерпретацію методу штрафних функцій з квадратичною штрафною функцією такого вигляду:

$$P_k(x) = k \sum_{s=1}^p [\max\{g_s(x), 0\}]^2.$$

Припустимо, $f(x)$ – лінійна функція, що задана на відрізку $[a, b]$. Тоді обмеження задачі мають вигляд: $g_1(x) = a - x \leq 0$, $g_2(x) = x - b \leq 0$, а штрафна функція $P_k(x) = k([\max\{a - x, 0\}]^2 + [\max\{x - b, 0\}]^2)$.

На рис. 14.4 зображено функції $P_k(x)$ і $f_k(x)$ для різних значень параметра k : $k_1 = 1, k_2 = 4$. $x_{k_1}^*$ і $x_{k_2}^*$ є точками мінімуму функцій $f_1(x)$ і $f_4(x)$ відповідно, а $x^* = a$ – розв'язок вихідної задачі. З рис. 14.4 видно, що послідовність точок мінімуму розширених функцій $f_k(x)$ розташована поза

допустимою областю (зовні відрізка $[a, b]$), але із зростанням параметра k прямує до точки мінімуму вихідної функції.

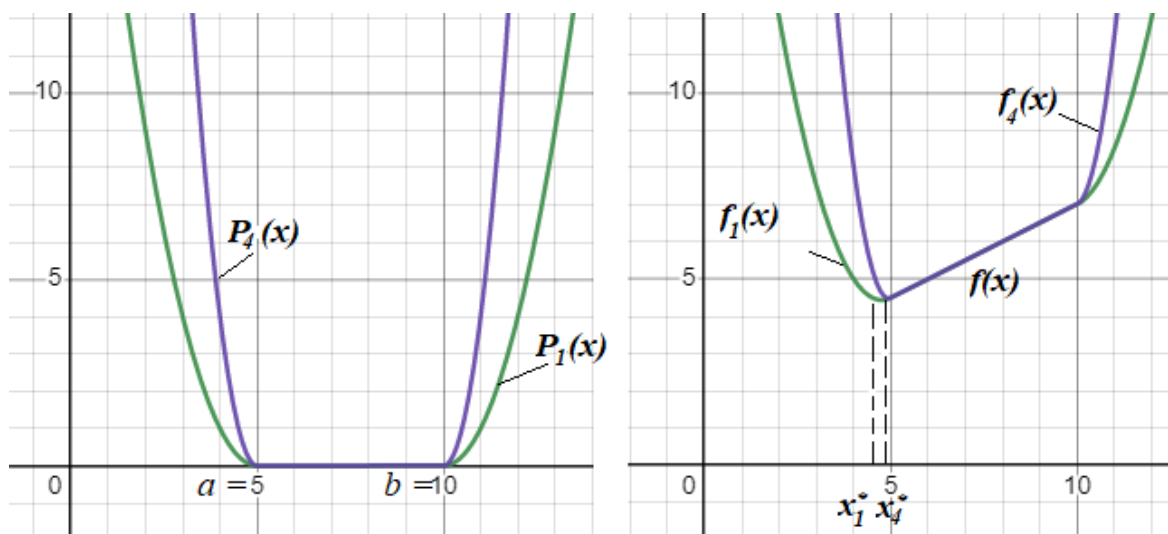


Рис. 14.4. Ілюстрація методу зовнішньої точки

Методи штрафних функцій такого типу отримали назву методів зовнішньої точки. Вони апроксимують функцію $P_\infty(x)$ функцією $P_k(x)$, яка поза допустимою областю наближується до $P_\infty(x)$, коли $k \rightarrow \infty$. Відтак, $P_k(x)$ є «штрафом» за вихід за межі допустимої області.

Дамо загальний опис методів зовнішньої точки. Розглянемо штрафну функцію у вигляді $P_k(x) = S(k)Q(x)$, де функції $S(k)$ і $Q(x)$ характеризуються наступними властивостями:

- 1) якщо $0 < k_1 < k_2$, то $0 < S(k_1) < S(k_2)$;
- 2) якщо $\{k_i\}$ – монотонно зростаюча послідовність додатних чисел і $\lim_{i \rightarrow \infty} k_i = \infty$, то $\lim_{i \rightarrow \infty} S(k_i) = \infty$;
- 3) $Q(x)$ – неперервна функція від x і $Q(x) = 0$, якщо $g_i(x) \leq 0$ (тобто якщо $x \in \Omega$) і $Q(x) > 0$ у протилежному випадку.

Тоді розширену функцію запишемо у такому вигляді:

$$f_k(x) = f(x) + S(k)Q(x).$$

Функції (14.3), (14.4), (14.6) можуть слугувати прикладами штрафних функцій, які використовуються в методах зовнішніх точок, коли $S(k) = k$.

Під час вибору штрафної функції, яка задовольняє умови 1) – 3), коли точка x знаходиться занадто далеко від допустимої області, то із зростанням k значення штрафної функції $P_k(x) = S(k)Q(x)$ зростає, тобто якщо $k_i \rightarrow \infty$, то $P_{k_i}(x) \rightarrow \infty$, при цьому точки мінімуму $x_{k_i}^*$ розширених функцій повинні наближатися до допустимої області, аби мінімізувати значення штрафної функції. Звідси випливає, що рух здійснюється зовні допустимої області у допустиму. З цієї причини методи штрафних функцій і називаються методами зовнішньої точки.

Обґрунтування методів зовнішньої точки дається теоремою 14.1.

Наведемо алгоритм розв'язання задачі (14.2) методом зовнішньої точки.

1. Для деякого $k_1 > 0$ за допомогою будь-якого відомого методу безумовної оптимізації визначається локальний мінімум функції $f_{k_1}(x)$. Позначимо його $x_{k_1}^*$;
2. Якщо вже відомий $x_{k_1}^*$, то за допомогою того самого методу знаходимо локальний мінімум $x_{k_2}^*$ функції $f_{k_2}(x)$ при $k_2 > k_1$;
3. Процес повторюється для строго зростаючої послідовності $\{k_i\}$.

Згідно з теоремою 14.1, коли $k_i \rightarrow \infty$, послідовність $\{x_{k_i}^*\}$ прямує до умовного локального мінімуму x^* задачі (14.2). Отже, процес побудови послідовності $x_{k_i}^*$ на деякому кроці можна обірвати і останній з отриманих мінімумів $x_{k_i}^*$ прийняти за наближений розв'язок задачі (14.2).

Приклад 14.3. Розглянемо задачу з прикладу 14.1. Мінімізувати функцію $f(x) = x^2 - 2$, коли $x \leq 1$.

Скориставшись функцією штрафу вигляду (14.3), запишемо розширену функцію, тобто:

$$f_k(x) = x^2 - 2 + k[\max\{x - 1, 0\}]^2$$

і розв'яжемо рівняння

$$\frac{df_k(x)}{dx} = 0.$$

$$\frac{df_k(x)}{dx} = 2x - 10 + 2k \cdot \max\{x - 1, 0\} = 0,$$

або

$$\max\{2x - 10 + 2k \cdot (x - 1), 2x - 10\} = 0.$$

Якщо $x > 1$, то $2x - 10 + 2k \cdot (x - 1) = 0$, звідки $x = \frac{\kappa+5}{\kappa+1}$. Коли ж $x \leq 1$, то $2x - 10 = 0$, але останнє рівняння задовольняє $x = 5 > 1$. Тому ясно, що єдиним розв'язком рівняння $\frac{df_k(x)}{dx} = 0$ є точка $x_k^* = \frac{\kappa+5}{\kappa+1}$.

Зрозуміло також, що за будь-якого $k > 0$ точка x_k^* перебуває поза допустимою областю і, коли $k \rightarrow \infty$, послідовність прямує до $x^* = 1$.

Метод бар'єрних функцій

Він застосовується для розв'язання задач умовної оптимізації з обмеженнями типу нерівностей, тобто:

$$\min f(x), x \in \Omega = \{x \in E^n, g_s(x) \leq 0, s = \overline{1, p}\}.$$

Зведемо задачу до задачі безумовної мінімізації функції $F(x, \beta_k)$.

$$F(x, \beta_k) = f(x) + P(x, \beta_k), x \in E^n.$$

Додаткова функція $P(x, \beta_k)$ вибирається таким чином, аби вона необмежено зростала, коли точка x прямує до межі області Ω .

Розглянемо штрафну функцію такого вигляду:

$$P(x, \beta_k) = -\beta_k \sum_{s=1}^p \frac{1}{g_s(x)},$$

де $x \in \Omega^{(0)}$, $\Omega^{(0)}$ – множина таких точок з Ω , для яких $g_s(x) < 0, s = \overline{1, p}$.

Очевидно, що $P(x, \beta_k)$ прямує до нуля коли $k \rightarrow \infty$ для $x \in \Omega^{(0)}$, а на межі області Ω кожна з функцій $P(x, \beta_k)$ нескінченно зростає.

Так само, як і у попередньому пункті, розглянемо лінійну функцію $f(x)$ на відрізку $[a, b]$. Штрафна функція $P(x, \beta_k)$ у цьому випадку буде записана у такий спосіб:

$$P_k(x) = -\frac{1}{k} \left(\frac{1}{x-b} - \frac{1}{x-a} \right).$$

Тут $\beta_k = 1/k$. На рис. 14.5 зображені функції $P_k(x)$ і $f_k(x)$ за різних значень параметра k : $k_1 = 1, k_2 = 4$.

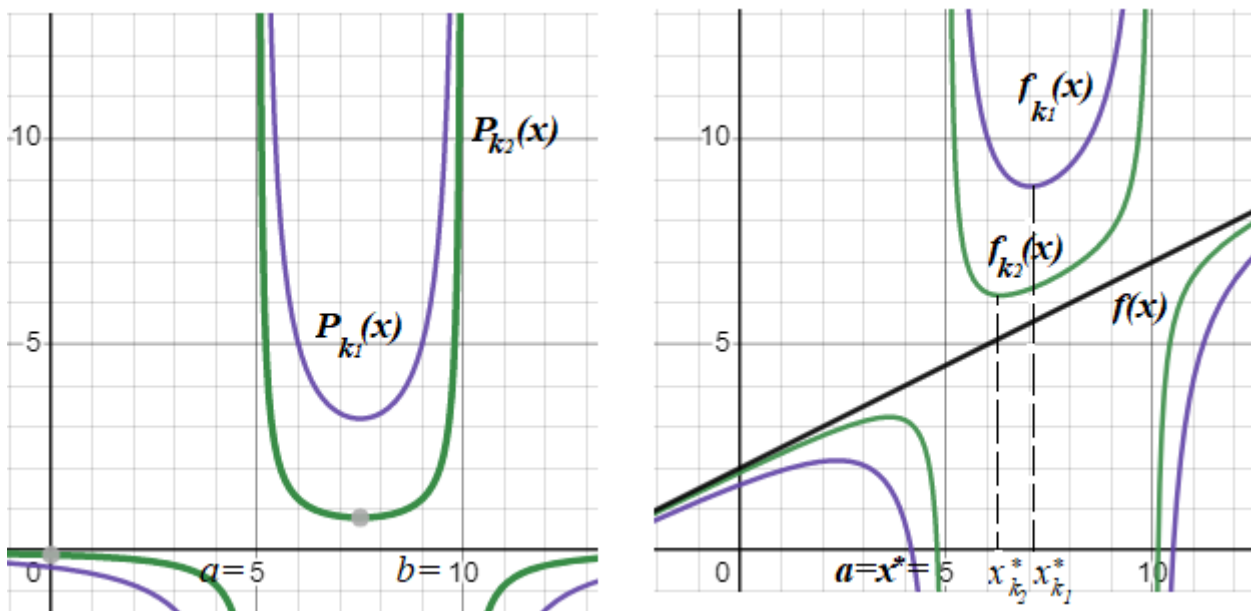


Рис. 14.5 Бар'єрні розширені функції в методі внутрішньої точки

Точки $x_{k_1}^*$ і $x_{k_2}^*$ є точками мінімуму функцій $f_1(x)$ і $f_4(x)$ відповідно, а $x^* = a$ – розв'язок вихідної задачі. З рис. 14.5 видно, що послідовність точок мінімуму розширених функцій $f_k(x)$ розташована всередині допустимої області (на відрізку $[a, b]$), але із зростанням параметра k прямує до точки мінімуму вихідної функції. Тому методи штрафних функцій такого типу отримали назву методів внутрішньої точки. Вони апроксимують функцію $P_\infty(x)$ функцією $P_k(x)$, яка зсередини допустимої області наближується до $P_\infty(x)$, коли $k \rightarrow \infty$.

Інакше кажучи, в цьому випадку штрафна функція ставить «бар'єр» проти виходу з допустимої області. Тому методи внутрішньої точки іноді називають методами бар'єрних функцій.

Дамо загальний опис методів внутрішньої точки.

Розглянемо штрафну функцію у такому вигляді:

$$P_k(x) = T(1/k)R(x),$$

де функції $T(1/k)$ і $R(x)$ характеризуються такими властивостями:

- 1) якщо $0 < k_1 < k_2$, то $0 < T(1/k_1) > T\left(\frac{1}{k_2}\right) > 0$;
- 2) якщо $\{k_i\}$ – монотонно зростаюча послідовність додатних чисел і $\lim_{i \rightarrow \infty} k_i = \infty$, то $\lim_{i \rightarrow \infty} T(1/k_i) = 0$;
- 3) $R(x)$ – неперервна функція в області $\Omega^{(0)} = \{x \in E^n, g_s(x) < 0, s = \overline{1, p}\}$.

Тоді розширеною функцією буде функція такого вигляду:

$$f_k(x) = f(x) + T(1/k)R(x).$$

Зауважимо, що функції (14.7), (14.8), (14.9) можуть слугувати прикладами штрафних функцій, які використовуються в методах зовнішніх точок, коли $T(1/k) = 1/k$.

Під час вибору штрафної функції, яка задовольняє умови 1) – 3), безумовний мінімум функцій $f_k(x)$ досягається всередині допустимої області, але не лежить на її границі за жодного значення $k > 0$, оскільки, наближаючись до границі допустимої множини з будь-якої її внутрішньої точки, функція $f_k(x)$ прямує до нескінченності. До того ж із зростанням k вплив штрафної функції зменшується і збільшується вплив цільової функції. Отже, коли $k_i \rightarrow \infty$, $x_{k_i}^* \rightarrow x^*$, залишаючись всередині допустимої множини.

За достатньо загальних припущень, аналогічних до теореми 14.1, можна надати строге обґрунтування методам внутрішньої точки. Суттєва відмінність є лише в тім, що у даному випадку вимагається непорожність внутрішності множини Ω .

Наведемо алгоритм розв'язання задачі (14.2) методом внутрішньої точки.

1) Для деякого $k_1 > 0$ будемо функцію $f_{k_1}(x)$; за початкове наближення для знаходження локального мінімуму $x_{k_1}^*$ функції $f_{k_1}(x)$ вибираємо деяку точку $x^{(0)} \in \Omega^{(0)}$;

2) за допомогою будь-якого відомого методу безумовної оптимізації визначаємо локальний мінімум $x_{k_1}^*$ функції $f_{k_1}(x)$. При цьому $x_{k_1}^* \in \Omega^{(0)}$;

3) вибираємо $x_{k_1}^*$ за початкове наближення і знаходимо локальний мінімум $x_{k_2}^*$ функції $f_{k_2}(x)$ при $k_2 > k_1 > 0$;

4) процес повторюємо для строго зростаючої послідовності $\{k_i\}$, тобто знаходимо локальний мінімум функції $f_{k_i}(x)$, виходячи з $x_{k_{i-1}}^*$. При $k_i \rightarrow \infty$ послідовність $\{x_{k_i}^*\}$ прямує до умовного локального мінімуму x^* задачі (14.2). Отже, процес побудови $x_{k_i}^*$ на деякому кроці можна обірвати і останній з отриманих мінімумів $x_{k_i}^*$ можна взяти за наближений розв'язок задачі (14.2).

Відтак, суттєвим в даному методі є те, що початковий коефіцієнт штрафу β_0 задається великим. Початкова точка x^0 обирається лише з середини області Ω , тому метод називається методом внутрішньої точки. Коефіцієнт β_k зменшується з кожною ітерацією ($\beta_k \rightarrow 0, k \rightarrow \infty$). При цьому $x^*(\beta_k) \rightarrow x^*$.

Приклад 14.3. Знайти мінімум функції $f(x) = x_1 + x_2$ за умов

$$g_1(x) = x_1^2 - x_2 \leq 0, \quad g_2(x) = -x_1 \leq 0.$$

Виберемо

$$T\left(\frac{1}{k}\right) = \frac{1}{k}, \quad R(x) = \ln\left(-\frac{1}{x_1^2 - x_2}\right) + \ln\left(-\frac{1}{kx_1^2 - x_2}\right).$$

Тоді розширена функція набуде такого вигляду:

$$f_k(x) = x_1 + x_2 - \frac{1}{k} \ln(-x_1^2 + x_2) - \frac{1}{k} \ln(x_1).$$

Оскільки ця функція є диференційовною, то можна виписати необхідні умови мінімуму:

$$\frac{\partial f_k(x)}{\partial x_1} = 1 + \frac{2x_1}{k(-x_1^2 + x_2)} - \frac{1}{kx_1} = 0,$$

$$\frac{\partial f_k(x)}{\partial x_2} = 1 - \frac{1}{k(-x_1^2 + x_2)} = 0.$$

Розв'язуючи отриману систему рівнянь, знаходимо:

$$x_{1k}^* = \frac{-1 \pm \sqrt{1 + 8/k}}{4}.$$

Оскільки згідно з другим обмеженням $x_1 \geq 0$, то маємо

$$x_{1k}^* = \frac{-1 + \sqrt{1 + 8/k}}{4},$$

і тоді

$$x_{2k}^* = \frac{1}{k} + \frac{(-1 + \sqrt{1 + 8/k})^2}{16}.$$

Для перевірки достатніх умов мінімуму побудуємо матрицю других похідних функції $f_k(x)$ і визначимо її компоненти в стаціонарній точці:

$$\frac{\partial^2 f_k(x)}{\partial x_1^2} = \frac{2}{k(-x_1^2 + x_2)} + \frac{4x_1^2}{k(-x_1^2 + x_2)^2} + \frac{1}{kx_1^2},$$

$$\left. \frac{\partial^2 f_k(x)}{\partial x_1^2} \right|_{(x_{1k}^*, x_{2k}^*)} = 2 + 4kx_{1k}^{*2} + \frac{1}{kx_{1k}^{*2}},$$

$$\frac{\partial^2 f_k(x)}{\partial x_2^2} = \frac{1}{k(-x_1^2 + x_2)^2}; \quad \left. \frac{\partial^2 f_k(x)}{\partial x_2^2} \right|_{(x_{1k}^*, x_{2k}^*)} = \frac{1}{k};$$

$$\frac{\partial f_k(x)}{\partial x_1 \partial x_2} = -\frac{2x_1}{k} \cdot \frac{1}{(-x_1^2 + x_2)^2}; \quad \left. \frac{\partial^2 f_k(x)}{\partial x_1 \partial x_2} \right|_{(x_{1k}^*, x_{2k}^*)} = -2kx_{1k}^*.$$

Вочевидь, матриця других похідних функції $f_k(x)$ в точці (x_{1k}^*, x_{2k}^*) є додатно визначеною, тому ця точка є точкою мінімуму функції $f_k(x)$. В табл. 14.1 наведено значення x_{1k}^*, x_{2k}^* для чотирьох значень k .

Геометрична інтерпретація задачі наведена на рис. 6. Допустима область на рисунку заштрихована, всі наближення лежать всередині допустимої множини і, коли $k \rightarrow \infty$, прямують до точного розв'язку $x^* = (0,0)$.

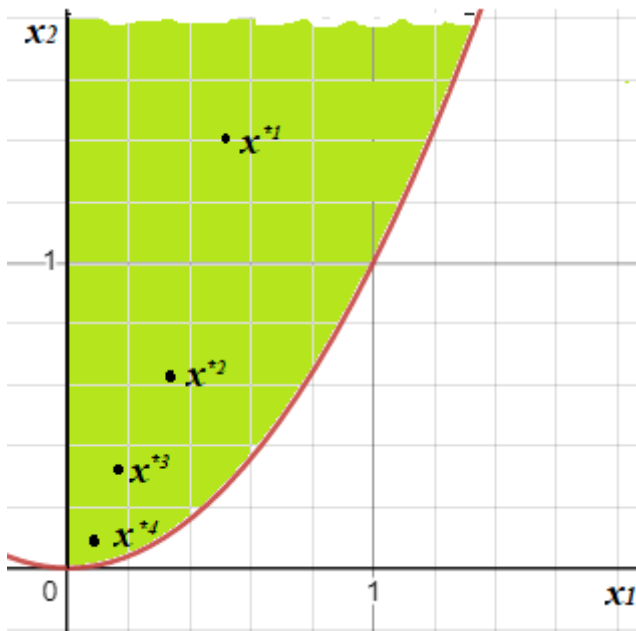


Рис. 6. Ілюстрація методу внутрішньої точки

Таблиця 14.1.
Оптимальні розв'язки задач мінімізації розширеної функції в прикладі 14.3

k	x_1	x_2
1	0.5	1.25
2	0.309017	0.595492
3	0.228714	0.385643
4	0.183013	0.283494
5	0.153113	0.223444

Метод штрафних функцій в задачах нелінійного програмування з обмеженнями у формі рівностей і нерівностей

Комбінований метод штрафних функцій

Методи внутрішньої та зовнішньої точок засновані на суттєво різних принципах. Як вже було зазначено, у методах внутрішньої точки штрафний доданок перешкоджає порушенню обмежень, тому всі наближення до оптимуму знаходяться всередині допустимої області. В методах зовнішньої точки, навпаки, у процесі розв'язування утворюється послідовність точок, яка виходить за межі допустимої області, штрафний доданок у цьому випадку запобігає блуканням точок надто далеко від допустимої області. Крім того, в методах внутрішньої точки за початкове наближення для знаходження мінімуму розширеної функції необхідно вибирати допустиму точку, в той час як у методах зовнішньої точки взагалі не потрібно вирішувати питання про допустимість точок послідовності. У цьому сенсі метод зовнішньої точки простіший. Однак практика показує, що в більшості випадків безумовна мінімізація функцій у методах зовнішньої точки є більш трудомісткою, ніж у методах внутрішніх точок.

Незважаючи на те, що збіжність обох алгоритмів залежить від одних і тих самих умов, за винятком умови, яка передбачає наявність внутрішньої допустимої області в методі внутрішньої точки, ситуації, в яких надається перевага тому чи іншому методу, ще не визначені. Слід зазначити, що для задач з обмеженнями у формі рівності методи внутрішньої точки не застосовні, оскільки вони потребують існування внутрішності допустимої області. Тому для розв'язання задач математичного програмування, котрі включають як обмеження-нерівності, так і обмеження-рівності, не можна уникнути застосування комбінованих методів.

Розглянемо загальну задачу математичного програмування:

$$\min f(x), x \in R = \{\varphi_j(x) = 0, j = \overline{1, m}, g_s(x) \leq 0, s = \overline{1, p}\}. \quad (14.10)$$

Розширену функцію для комбінованого алгоритму виберемо у такому вигляді:

$$f_k(x) = f(x) + T\left(\frac{1}{k}\right)R(x) + S(k)Q(x).$$

Припустимо, що функції T, R, S, Q мають описані вище властивості, причому R визначається обмеженнями-нерівностями, а Q – обмеженнями-рівностями.

Як розширені в комбінованих методах можуть виступати, наприклад, такі функції:

$$f_k(x) = f(x) - \frac{1}{k} \sum_{s=1}^p \frac{1}{g_s(x)} + \sqrt{k} \sum_{j=1}^m \varphi_j^2(x), \quad (14.11)$$

$$f_k(x) = f(x) - \frac{1}{k} \sum_{s=1}^p \ln(-g_s(x)) + k \sum_{j=1}^m \varphi_j^2(x). \quad (14.12)$$

Фіакко і Маккорміком був розроблений алгоритм послідовної безумовної мінімізації для розв'язування задачі нелінійного програмування вигляду (14.11), в якій функції $f(x)$ і $g_s(x)$ можуть бути нелінійними, а $\varphi_j(x)$ мають бути лінійними функціями незалежних змінних. У Фіакко і Маккорміка описано два варіанти методу послідовної безумовної мінімізації: перший – для розширеної функції вигляду (14.11), другий – для розширеної функції вигляду (14.12).

Опишемо схему обчислювального методу послідовної безумовної мінімізації

1. Знаходимо точку $x^{(0)} \in \Omega^{(0)} = \{x \in E^n, g_s(x) < 0, s = \overline{1, p}\}$, тобто початкова точка має бути внутрішньою для множини, яка визначена обмеженнями типу нерівностей. Ця вимога є суттєвою. Оскільки розширена функція може мати кілька мінімумів, то, розпочинаючи процес з точки, яка не є внутрішньою, можна прийти до мінімуму, котрий не буде допустимим.

Якщо точку відразу знайти нелегко, можна застосувати спеціальні методи, про які йтиметься нижче.

2. Визначаємо $k_1 > 0$ – початкове значення параметра k .

3. Знаходимо безумовний мінімум функції $f_k(x)$ для поточного значення $k = k_i$, цей крок становить основний обсяг роботи.

4. Перевіряємо критерії закінчення обчислювального процесу. Якщо вони виконані, то процес закінчуємо, в іншому випадку переходимо до кроку 5.

5. Вибираємо k_{i+1} . Переходимо до кроку 3.

Для практичної реалізації описаного вище алгоритму необхідно з'ясувати такі питання:

а) за яких умов метод послідовної безумовної мінімізації є збіжним?

б) як знаходити початкову внутрішню точку?

в) якими способами можна визначати параметр k ?

г) за допомогою яких методів можна мінімізувати розширену функцію?

д) якими є критерії закінчення ітераційного процесу?

Умови збіжності методу послідовної безумовної мінімізації

Розглянемо питання про збіжність комбінованого методу штрафних функцій для випадку, коли функції $f(x)$, $\varphi_j(x)$, $j = \overline{1, m}$, $g_s(x)$, $s = \overline{1, p} \in$ двічі неперервно диференційовними. Тоді для мінімізації розширених функцій можна застосовувати методи, засновані на використанні перших і других похідних.

Припустимо, що виконані такі умови:

1. Задача (14.10) має непорожню допустиму область.

2. Для будь-якого скінченного C і будь-якого $k > 0$ множина точок, що належать множині Ω і задовольняють нерівність

$$f(x) + \sqrt{k} \sum_{j=1}^m \varphi_j^2(x) \leq 0,$$

є обмеженою.

3. Цільова функція $f(x)$ і сума $\sum_{j=1}^m \varphi_j^2(x)$ є опуклими.

4. Функції $g_s(x)$, $s = \overline{1, p}$ опуклі.

5. Матриця других частинних похідних розширеної функції $f_k(x)$ не дорівнює нулю в жодній з точок, що належать множині Ω .

Умови 1 – 5 є достатніми для збіжності методу послідовної безумовної мінімізації. Фіакко і Маккормік довели наступну теорему про збіжність цього методу, спираючись на умови 1 – 5.

Теорема 14.2. Якщо задача нелінійного програмування задовольняє умови 1 – 5, то:

1) кожна розширена функція $f_k(x)$ має мінімум в деякій точці x_k^* множини Ω ;

2) якщо послідовність $\{k_i\}$ є строго монотонно зростаючою, то

$$\lim_{i \rightarrow \infty} f_{k_i}(x) = \min_{x \in \Omega} f(x) = f(x^*) = f^*,$$

тобто послідовність безумовних мінімумів функцій $f_k(x)$ прямує до умовного мінімуму функції $f(x)$.

Примітка. Для задачі нелінійного програмування з обмеженнями у формі рівностей і нерівностей можна застосовувати метод зовнішньої точки. При цьому штрафною функцією може бути така:

$$P(x, \beta_k) = \beta_k \left[\sum_{j=1}^m \varphi_j^2(x) + \sum_{s=1}^p \hat{g}_s^2(x) \right]; \quad \hat{g}_s(x) = \begin{cases} g_s(x), & g_s(x) > 0, \\ 0, & g_s(x) \leq 0. \end{cases}$$

Початкове наближення x^0 вибирають поза допустимою областю Ω , тому цей метод є методом зовнішньої точки. Суттєвим в даному випадку є те, що початковий коефіцієнт штрафу β_0 задають невеликим, аби зменшити «яружність» розширеної функції $F(x, \beta_k)$. Далі β_k зростає з кожною ітерацією, а $x^*(\beta_k) \rightarrow x^*$, коли $k \rightarrow \infty$.

Вибір початкового наближення. Початкову точку x в методах бар'єрних функцій потрібно вибирати так, щоб задовольнити умови $g_s(x) < 0$, тобто x має бути внутрішньою точкою множини, яка визначена обмеженнями-нерівностями. Якщо таку точку знайти складно, то можна діяти таким чином:

- 1) вибрати точку, яка задовольняє деяким з умов $g_s(x) < 0$ (за можливістю найбільшій їх кількості);
- 2) зафіксувати перше порушене обмеження-нерівність та взяти функцію, що відповідає цьому обмеженню, за тимчасову цільову;
- 3) мінімізувати цю функцію за допомогою методу послідовної безумовної мінімізації, взявши до уваги враховані на поточний момент умови;
- 4) як тільки виділена обмежувальна умова виявиться задоволеною, її слід перевести в розряд звичайних обмежень і повторити процедуру для іншого порушеного обмеження, тобто зафіксувати іншу тимчасову цільову функцію і т.д.

Для визначення потрібної внутрішньої точки можна також мінімізувати суму всіх тих $g_s(x)$, для яких відповідні обмеження виявляються порушеними.

Методика визначення параметра k . Існують різні способи вибору початкового значення k_1 для параметра k . Вибираючи k , слід враховувати, що за малих його значень процес мінімізації починається у внутрішніх точках, що лежать занадто далеко від границі допустимої області, тоді як занадто великі значення k призводять до того, що початкове наближення точки мінімуму виявляється надмірно близьким до границі описаної обмеженнями допустимої множини. В обох випадках для того, щоб точка опинилася в потрібному місці допустимої області, доводиться витратити додатковий час. З практичної точки зору найбільш зручно вважати, що k дорівнює одиниці.

Що ж до вибору подальших значень параметра, то, як показує практика, алгоритм пошуку мінімуму в задачі умовної оптимізації (4.10) методом штрафних функцій буде досить ефективним, якщо послідовність k_1, k_2, k_3, \dots індукувати таким простим співвідношенням: $k_{i+1} = k_i \cdot C$, де $C > 1$ є константою. Часто передбачають, що $C = 4$.

Очевидно, що зі збільшенням значення k ми отримуємо все більш точне значення наближеного розв'язку, проте необхідно зазначити, що оперування з великими числами k вносить додаткові труднощі в обчислювальний процес і викликає додаткову похибку обчислень. Крім того, як показує аналіз оцінок похибок багатьох методів безумовної мінімізації, швидкість збіжності цих методів уповільнюється за великих значень k , тому значення параметра k слід збільшувати доти, поки вдається отримувати досить швидке зменшення функції $f(x)$ з невеликою витратою машинного часу. При цьому для малих k занадто точне розв'язання допоміжної задачі недоцільне, необхідно лише дбати про те, аби із зростанням k задача розв'язувалася все точніше. Якщо вдається таким чином отримати розв'язок з необхідної точністю, і процес руху до мінімуму вихідної задачі сповільнюється, то далі вдаються до залучення інших методів мінімізації.

Мінімізація розширених функцій. Основну складність під час реалізації методів штрафних функцій становить мінімізація розширеної функції $P_k(x)$ на всьому просторі E_n . Тому, насамперед вибір методу мінімізації $P_k(x)$ має бути узгоджений із вибором штрафної функції. Якщо функція є неперервною диференційовною (ця умова виконується, наприклад, у разі вибору штрафної

функції за однією з формул (14.3) – (14.5), (14.7) – (14.9) і обчислення її похідних не викликає значних труднощів, то для безумовної мінімізації можна застосовувати традиційні методи найшвидшого спуску та сполучених градієнтів. Останній особливо ефективний, якщо розширена функція є квадратичною, оскільки в цьому випадку метод збігається за кінцеве число кроків. Можна також застосовувати різні методи змінної метрики.

Коли ж розширена функція є двічі неперервно диференційовною, то для її мінімізації природно застосовувати метод Ньютона. Незважаючи на труднощі, пов'язані з обчислювальною реалізацією цього методу (обчислення на кожному кроці матриці зворотної до гессіана, великий обсяг пам'яті комп'ютера та інше) він виявляється набагато ефективнішим за інші.

Вибір негладких функції штрафу, наприклад (14.6), спричиняє необхідність мінімізації розширеної негладкої функції. Існує низка відомих методів розв'язання таких задач: метод узагальненого градієнтного спуску; методи, обчислювальні схеми яких не містять похідних, а використовують тільки значення функції. Сюди відносять метод прямого пошуку, метод пошуку за деформованим багатогранником, метод Пауелла, випадкового пошуку та інші. Слід зазначити, що ситуації, в яких похідні функції не існують або не можуть бути обчислені явно, часто виникають при розв'язуванні практичних задач, наприклад, коли значення функції являють собою результати експерименту. У багатьох випадках, коли похідна існує, але її аналітичний вираз надзвичайно громіздкий, також залучають методи, які працюють лише зі значеннями функції.

Задачі безумовної мінімізації розширених функцій несуть у собі відбиток тих труднощів, які притаманні методам безумовної мінімізації, і у практичній реалізації методів штрафних функції ці проблеми можуть бути дуже значними. Тому важливою є робота зі створення нових ефективних методів безумовної мінімізації. На сьогодні ефективними є модифікований метод Ньютона і модифікований метод змінної метрики для мінімізації розширених функцій за наявності похідних.

Про критерії закінчення ітераційного процесу. Найбільше значення параметра k , для якого ще слід шукати безумовний мінімум розширеної функції, залежить суттєво від обраних критеріїв збіжності.

Природний критерій для задач опуклого програмування дає теорія двоїстості, але для того, щоб його використовувати, потрібно поряд з вихідною задачею розв'язувати двоїсту до неї.

На практиці як критерій закінчення ітераційного процесу в методі послідовної безумовної мінімізації часто застосовується наступна умова:

$$\left| \frac{1}{k} \sum_{s=1}^p \frac{1}{g_s(x)} \right| \leq \varepsilon,$$

тут $\varepsilon > 0$ – деяка константа. Але ця умова не містить члена, який дозволив би перевірити, чи задовольняє поточна точка локального мінімуму умови рівності: $\varphi_j(x) = 0, j = \overline{1, m}$.

За критерій закінчення ітераційного процесу методу штрафних функцій можна також прийняти близькість мінімальних значень цільової функції для двох послідовних значень параметра k .

4. Метод модифікованих функцій Лагранжа

Найбільш перспективним загальним методом врахування обмежень вважається метод модифікованих функцій Лагранжа.

Розглянемо його стосовно такої задачі:

$$f(x) \rightarrow \min, x \in \Omega,$$

$$\Omega = \{x \in \mathbb{R}^n: g_i(x) = 0, 1 \leq i \leq s\}.$$

Візьмемо за узагальнений критерій оптимальності функцію

$$M(x, \lambda, \sigma) = f(x) + \sum_{i=1}^s \lambda_i g_i(x) + \frac{\sigma}{2} \sum_{i=1}^s g_i^2(x),$$

де $\sigma > 0$ – параметр методу.

Тоді алгоритм оптимізації зводиться до ітераційного процесу

$$x^{k+1} = \arg \min_x M(x, \lambda^k, \sigma),$$

$$\lambda^{k+1} = \lambda^k + \sigma g(x^{k+1}); g(x^{k+1}) = [g_1(x), g_2(x), \dots, g_s(x)],$$

який узагальнює методи штрафних функцій і множників Лагранжа.

Особливістю сформульованого алгоритму у порівнянні з методом штрафних функцій є відсутність нескінченно зростаючого штрафного коефіцієнта σ . За умови правильної організації процесу величина σ впливає лише на швидкість збіжності, а не на самий факт збіжності послідовності $\{x^k\}$ до оптимуму. Зауважимо, що під час розв'язання практичних задач значення σ доцільно підбирати в інтерактивному режимі, оскільки надійні методи апріорного задання σ сьогодні відсутні.

Задачі для розв'язування у класі

Визначити, який метод можна застосовувати до розв'язання поданих нижче задач умовної оптимізації і виконати 2 – 3 ітерації цього методу.

1. $f = 4(x_1 + 1)^2 + (x_2 - 14)^2 \rightarrow \text{extr},$ $x_1 + x_2 - 1 = 0.$
2. $f = (x_1 + 2)^2 + 4(x_2 - 1)^2 \rightarrow \text{extr},$ $(x_1 - 4)^2 + (x_2 + 1)^2 \leq 9.$

$$\begin{array}{ll}
3. f = 4/x_1 + 9/x_2 + x_1 + x_2 \rightarrow \min, & 4. z = -x_1^2 - 2x_2^2 + 2x_1 + 12x_2 \rightarrow \text{extr}, \\
4x_1 + 5x_2 \leq 25, & 4x_1 + 5x_2 \leq 40, \\
x_j \geq 0, j = \overline{1,2}. & x_1 - 5x_2 = 10.
\end{array}$$

Рекомендована література та інформаційні ресурси

1. Нефьодов Ю.М. Методи оптимізації в прикладах і задачах : навчальний посібник / Ю.М. Нефьодов, Т.Ю. Балицька. – Київ : Кондор, 2011. – 324 с.
2. Оптимізація систем керування.: навч. посіб. для студ. спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» / Л. Р. Ладієва; КПІ ім. Ігоря Сікорського, 2020. – 211 с.

Контрольні питання

1. Наведіть приклади заміни змінних, які дозволяють уникнути умови невід'ємності на вихідні змінні.
2. Сформулюйте основну ідею методу штрафних функцій.
3. Коли використовуються функції зовнішнього штрафу, бар'єрні функції?
4. Який сенс мають бар'єрні функції?
5. Яка основна ідея методу модифікованих функцій Лагранжа?
6. Для розв'язання яких задач оптимізації можна застосовувати комбінований метод штрафних функцій?
7. Чому методи штрафних функцій називаються параметричними?
8. Які методи можна застосовувати до розв'язання задач безумовної мінімізації розширених функцій?
9. Як задавати або змінювати параметр під час програмної реалізації методу штрафних функцій?

Завдання для індивідуальної роботи

Використовуючи дані із індивідуального завдання з Практичного заняття № 2, реалізувати метод зовнішнього штрафу (якщо номер варіанта непарний) або метод бар'єрних функцій (коли номер варіанта парний), і метод модифікованих функцій Лагранжа. Виконати порівняльний аналіз отриманих результатів. Порівняти їх із результатами графічного способу розв'язання.

Практичне заняття № 15

Евристичні алгоритми оптимізації

Мета: ознайомитися з ідеями евристичних алгоритмів оптимізації і особливостями їх застосування на практиці

План заняття

1. Огляд евристичних методів оптимізації: алгоритми імітації відпалу, стрибку по басейну, рою частинок;
2. Метод диференціальної еволюції;
3. Генетичний алгоритм і його модифікації.

Опорні поняття

1. Огляд евристичних методів оптимізації

В інформатиці та математичній оптимізації метаевристика – це процедура вищого рівня, яка має на меті забезпечити достатньо гарний розв'язок задачі оптимізації, особливо з неповною або недосконалою інформацією або в умовах обмежених обчислювальних можливостей. Метаевристика відбирає підмножину розв'язків, яка в іншому випадку є занадто великою, аби бути повністю дослідженою іншим способом. Вона потребує порівняно мало припущень щодо вирішуваної задачі оптимізації, і тому може бути придатною для різноманітних проблем.

Порівняно з алгоритмами оптимізації та ітераційними методами, метаевристика не гарантує, що для певного класу проблем можна знайти глобальний оптимальний розв'язок. Багато метаевристичних алгоритмів реалізують певну форму стохастичної оптимізації, і тому знайдений розв'язок залежить від набору згенерованих випадкових величин.

Дуже активним напрямком досліджень сьогодні є розробка еволюційних алгоритмів, заснованих на обчисленнях, що відтворюють природні явища. Природа виступає джерелом концепцій, механізмів та принципів для проектування штучних обчислювальних систем і для розв'язання складних обчислювальних проблем.

Алгоритм імітації відпалу та його модифікації.

Назва даного алгоритму пов'язана з методами імітаційного моделювання в статистичній фізиці, заснованими на техніці Монте-Карло. Дослідження кристалічної решітки і поведінки атомів при повільному охолодженні тіла спричинило розробку ймовірнісних алгоритмів, які виявилися надзвичайно ефективними в комбінаторній оптимізації. Вперше це було помічено у 1983 році. Окрім цього, існує обґрунтований інтерес до можливості застосування цього алгоритму в задачах неперервної оптимізації.

Сьогодні цей алгоритм є популярним як серед практиків завдяки своїй простоті, гнучкості і ефективності, так і серед теоретиків, оскільки вдається аналітично досліджувати його властивості і довести асимптотичну збіжність. Тим не менш, незважаючи на широку сферу застосування, швидкість збіжності методу відпалу все ще мало вивчена.

Метод відпалу використовується для пошуку глобального мінімуму деякої функції $f(x)$, заданої для x з деякого простору S , дискретного або неперервного. Елементи множини S являють собою стани уявної фізичної системи (енергетичні рівні), а значення функції f в цих точках розглядається як енергія системи $E = f(x)$. В кожен момент часу передбачається, що задано температуру системи T , як правило, вона зменшується з плином часу. Після потрапляння в стан x при температурі T наступний стан системи вибирається відповідно до заданого *породжувального сімейства ймовірнісних розподілів* $G(x, T)$, яке при фіксованих x і T задає випадковий елемент $G(x, T)$ зі значеннями в просторі S . Після генерації нового стану $x' = G(x, T)$, система з імовірністю $h(E, T)$ переходить до наступного кроку в стан x' , в іншому випадку процес генерації x' повторюється. Тут E позначає приріст функції енергії $f(x') - f(x)$. Величина $h(E, T)$ називається ймовірністю прийняття нового стану.

Як правило, за функцію $h(E, T)$ обирають або точне значення відповідної фізичної величини:

$$h(\Delta E, T) = \frac{1}{1 + \exp(\Delta E/T)}$$

або наближене її значення

$$h(\Delta E, T) = \exp\left(\frac{-\Delta E}{T}\right).$$

Зауважимо, що друга формула використовується частіше. При її застосуванні $h(E, T)$ виявляється більше одиниці, коли $E < 0$, і тоді вважають, що відповідна ймовірність дорівнює 1. Отже, якщо новий стан дає краще значення цільової функції, то перехід в цей стан станеться в будь-якому випадку.

Конкретна схема методу відпалу задається наступними параметрами:

- Вибором закону зміни температури $T(k)$, де k номер кроку.
- Вибором породжувального сімейства розподілів $G(x, T)$.
- Вибором функції ймовірності прийняття $h(E, T)$.

Алгоритм методу

1) Випадково обирається початкова точка $x = x_0$, $x_0 \in S$. Поточне значення енергії E обчислюють як $f(x_0)$.

2) k -та ітерація основного циклу складається з таких кроків:

- a) Порівняти енергію системи E в стані x із знайденим на поточний момент глобальним мінімумом. Якщо $E = f(x)$ менше, то змінити значення глобального мінімуму на поточну величину енергії системи.
- b) Згенерувати нову точку $x' = G(x, T(k))$.
- c) Обчислити значення функції в ній $E' = f(x')$.
- d) Згенерувати випадкове число α з інтервалу $[0; 1]$.
- e) Якщо $\alpha < h(E' - E, T(k))$, то встановити $x \leftarrow x', E \leftarrow E'$ і перейти до наступної ітерації. Інакше повторити крок b), поки не буде знайдена відповідна точка x' .

Основним недоліком описаного відпалу є дуже повільне спадання температури. Існує алгоритм, який дозволяє використовувати для зміни температури таку схему:

$$T(k) = \frac{T_0}{k} \quad (15.1)$$

без втрати гарантії знаходження глобального мінімуму.

Це досягається за рахунок використання сімейства відповідним чином нормованих розподілів Коші з щільністю

$$g(x'; x, T) = \frac{T}{(|x' - x|^2 + T^2)^{(D+1)/2}}$$

Наприклад, коли $D = 1$, приходимо до такої щільності:

$$g(x'; x, T) = \frac{1}{\pi} \frac{T}{|x' - x|^2 + T^2}$$

На жаль, цей розподіл не дуже зручно моделювати в просторі розмірності, більшої за 1. Цього можна уникнути, наприклад, за допомогою множення D одновимірних розподілів Коші, а саме:

$$g(x'; x, T) = \frac{1}{\pi^D} \prod_{i=1}^D \frac{T}{|x'_i - x_i|^2 + T^2}$$

але в цьому випадку знаходження глобального мінімуму гарантується, тільки коли закон зміни температури не швидше ніж $T(k) = \frac{T_0}{k^D}$, що набагато повільніше схеми (15.1).

Іншою модифікацією методу є схема подвійного відпалу. Цей стохастичний підхід поєднує узагальнення класичної імітації відпалу та швидкої імітації відпалу із стратегією застосування локального пошуку у певних областях. Він демонструє метод для вдосконалення розв'язку, знайденого узагальненим процесом відпалу. Модифікований алгоритм використовує деформований розподіл Коші – Лоренца, а його форма контролюється параметром q_v , тобто

$$g_{q_v}(\Delta x(t)) \propto \frac{[T_{q_v}(t)]^{-\frac{D}{3-q_v}}}{\left[1 + (q_v - 1) \frac{(\Delta x(t))^2}{[T_{q_v}(t)]^{3-q_v}}\right]^{\frac{1}{q_v-1} + \frac{D-1}{2}}}$$

де t – штучний час. Цей розподіл використовується для створення пробної дистанції стрибка $\Delta x(t)$ змінної $x(t)$ штучної температури $T_{q_v}(t)$.

З початкової точки після виклику функції розподілу ймовірність прийняття обчислюється у такий спосіб:

$$p_{q_a} = \min \left\{ 1, [1 - (1 - q_a)\beta\Delta E]^{-\frac{1}{1-q_a}} \right\},$$

де q_a – параметр прийнятності.

Для $q_a < 1$ нульова ймовірність прийняття призначається тим випадкам, коли

$$[1 - (1 - q_a)\beta\Delta E] < 0.$$

Штучна температура $T_{q_v}(t)$ знижується відповідно до

$$T_{q_v}(t) = T_{q_v}(1) \frac{2^{q_v-1} - 1}{(1+t)^{q_v-1} - 1},$$

де q_v параметр розподілу.

Алгоритм стрибка в басейні

Алгоритм стрибка в басейні – це стохастична глобальна оптимізація, яка була мотивована проблемами фізичної хімії, зокрема, задачами пошуку стабільних молекулярних конфігурацій або конфігурацій з найменшою енергією. Такі задачі зазвичай мають багато локальних мінімумів, що ускладнює використання стандартних методів оптимізації, оскільки існує залежність від початкових умов.

Стрибок в басейні як стохастичний алгоритм глобальної оптимізації використовує кілька концепцій, які успішно комбінуються. Перш за все, це підхід Монте-Карло, який застосовувався до подібних проблем і раніше. Другою складовою є локальний оптимізатор, який повертає для заданої вихідної точки локальний мінімум.

Алгоритм методу

1. Обирають вихідну точку.
2. Обчислюють локальний мінімум.
3. Проводять випадкове збурення координат локального мінімуму.
4. Обчислюють наступний локальний мінімум.

5. Порівнюють локальні мінімуми і зберігають найкращий.

Стохастична компонента існує для виконання випадкових переміщень або збурень у даній точці, яка має бути локальним мінімумом. Збурення, очевидно, повинно бути достатньо великим, щоб уникнути поточного локального мінімуму, але не надто великим, щоб стрибок не став абсолютно випадковим. Мотивація фізичної хімії полягає в тому, що стабільні конфігурації – локальні мінімуми – розташовані в одній і тій самій області пошукового простору, причому алгоритму не потрібно досліджувати весь пошуковий простір, що є однією з основних проблем глобальної оптимізації взагалі. Тобто, якщо мінімуми, які цікавлять, розташовані усі разом в якійсь області простору, то алгоритм стрибка в басейні може бути дуже ефективним.

На рис. 15.1 показано принципову схему, яка ілюструє ефекти перетворення енергії для одновимірного прикладу. Це просте, але потужне перетворення енергії, приєднане до певної конфігурації, що являє собою цільову функцію, яку слід мінімізувати у кусково-крокову функцію.

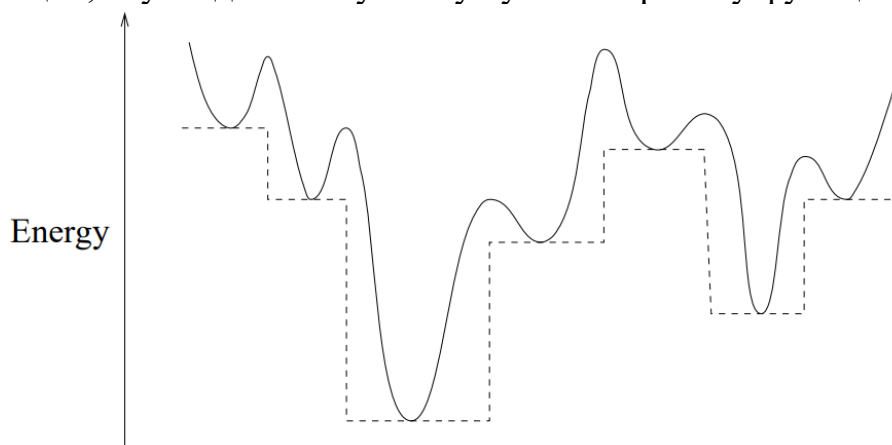


Рис. 15.1. Принципова схема ілюстрації ефекту перетворення енергії для одновимірного випадку. Суцільна лінія – енергія вихідної поверхні, пунктирна – перетворена енергія

При такому складному формулюванні задачі природно застосовувати певні наближення до проблеми, яка зазвичай означає застосування певного перетворення або послаблення. Ця трансформація має чудову властивість зберігати глобальні та локальні мінімуми. Крім того, вона ілюструє термін "стрибок в басейні" та деякі труднощі, пов'язані з цим – ефективність стрибка залежить від ширини басейну відповідно до ступеня, який контролює випадкові збурення.

Метод рою частинок

Метод оптимізації за допомогою рою частинок, який базується на моделюванні поведінки популяції частинок в просторі параметрів задачі оптимізації, привабливий простотою реалізації і тим, що в процесі обчислення не використовується градієнт. Він може використовуватися для розв'язання багатьох задач, включаючи навчання нейромереж, завдання пошуку мінімуму функції, а також задач, типових для застосування генетичних алгоритмів.

Метод рою частинок, як і всі алгоритми, що належать до сімейства еволюційних алгоритмів, є стохастичним. Це не вимагає обчислення градієнта і дозволяє використовувати його у випадках, де обчислення градієнта неможливо або має високу обчислювальну складність.

Алгоритм використовує для розв'язання рій частинок, де кожна з них являє собою можливий розв'язок задачі оптимізації. Нехай s – кількість агентів в рої. Кожна i -а частинка може бути представлена як об'єкт з низкою параметрів, а саме: x_i – стан частинки; v_i – її швидкість; y_i – її найкраще особисте положення.

Найкраще особисте положення частинки – це її положення, якому відповідає найкраще значення цільової функції серед всіх, які вона коли-небудь відвідувала.

Припустимо, f – функція, яку необхідно мінімізувати, тоді вираз для найкращого особистого положення залежно від часу можна описати у такий спосіб:

$$y_i(t + 1) = \begin{cases} y_i(t), & \text{if } f(x_i(t + 1)) \geq f(y_i(t)), \\ x_i(t + 1), & \text{if } f(x_i(t + 1)) < f(y_i(t)). \end{cases}$$

Існує дві версії базового алгоритму рою частинок, які йменують $gbest$ і $lbest$. Різниця між ними полягає в тому, з яким набором частинок взаємодіє кожна частинка під час роботи алгоритму. Позначимо символом \hat{y} цю взаємодію. Визначення \hat{y} в разі $gbest$ подано в наступному виразі:

$$\hat{y} \in \{y_0(t), y_1(t) \dots y_s(t)\} \mid f(\hat{y}(t)) = \min\{f(y_0(t)), f(y_1(t)), \dots, f(y_s(t))\}.$$

Він говорить про те, що \hat{y} – найкраща позиція будь-коли відвідувана будь-якою частинкою з рою.

Стохастична складова алгоритму подана двома випадковими параметрами $r_1 \sim U(0,1)$ і $r_2 \sim U(0,1)$, які масштабуються за допомогою сталих коефіцієнтів прискорення c_1 і c_2 , що відповідають за величину кроку, який може зробити частинка за одну ітерацію часу. Як правило $c_1, c_2 \in (0,2]$. Швидкість частинки на i -му кроці розраховується окремо для кожного виміру $j \in 1 \dots n$, таким чином $v_{i,j}$ позначає j -й вимір вектора швидкості i -ої частинки. Розрахунок j -го компонента вектора швидкості i -ої частинки на $t + 1$ кроці здійснюється за такою формулою:

$$v_{i,j}(t + 1) = w_c v_{i,j}(t) + c_1 r_1(t) [y_{i,j}(t) - x_{i,j}(t)] + c_2 r_2(t) [\hat{y}(t) - x_{i,j}(t)].$$

Отже, c_2 керує впливом глобального кращого положення, а c_1 впливом особистого кращого положення на швидкість кожної частинки. Для поліпшення збіжності алгоритму вводиться коефіцієнт інерції w_c . Положення кожної частинки в i -му вимірі обчислюється за формулою:

$$x_i(t + 1) = x_i(t) + v_i(t + 1).$$

Загальна схема алгоритму

1. Створення рою частинок.
2. Знаходження найкращого розв'язку для кожної частинки.
3. Знаходження найкращого розв'язку серед усіх частинок.
4. Корекція швидкості кожної частинки.
5. Переміщення кожної частинки.
6. Якщо критерій зупинки виконаний (як правило, це перевищення наперед заданого числа ітерацій), то виводять результати, інакше потрібен перехід до пункту 2.

2. Метод диференціальної еволюції

Метод диференціальної еволюції відноситься до еволюційного моделювання. Його призначено для розв'язання задач багатовимірної оптимізації. За класифікацією оптимізаційних методів він є стохастичним, оскільки використовує в процесі пошуку рішення генератор випадкових чисел. Окрім цього, застосовує деякі ідеї генетичних алгоритмів, що робить їх дуже схожими, але, на відміну від канонічного генетичного алгоритму, він не вимагає роботи зі змінними в бінарному коді, або одразу спроектований для роботи з дійсними числами.

Метод диференціальної еволюції є прямим, тобто в ході його роботи потрібно тільки обчислювати значення цільової функції, але не її похідних. У загальному випадку, цільові функції, що оптимізуються за допомогою даного методу, можуть бути не диференційовні, нелінійні, багатоекстремальні і з дуже великою кількістю змінних. Він простий в реалізації і використанні і легко розпаралелюється.

Метод диференціальної еволюції, розроблений Рейнером Сторно і Кеннетом Прайсом, вперше опублікований в 1995 році і розвинений в їх більш пізніх роботах.

Алгоритм методу

1. Ініціалізують множину випадкових векторів, які називають поколінням і являють собою можливі розв'язки задачі оптимізації. Число векторів в кожному поколінні одне й те саме і є одним з параметрів налаштування методу.

2. На кожній епосі еволюційного процесу алгоритм генерує нове покоління векторів, випадковим чином комбінуючи між собою вектори попереднього покоління.

Генерація векторів нового покоління проводиться у такий спосіб: для кожного вектора \vec{x}_i зі старого покоління (базового вектора) вибирають три різних випадкових вектора $\vec{v}_1, \vec{v}_2, \vec{v}_3$, також серед векторів старого покоління, за винятком самого вектора \vec{x}_i , і генерують так званий мутантний вектор відповідно до співвідношення

$$\vec{v} = \vec{v}_1 + \varphi(\vec{v}_2 - \vec{v}_3),$$

де φ – один з параметрів налаштування методу. Він характеризує максимально можливу відстань, на яку може розширитись область пошуку оптимуму за однією змінною за одну епоху еволюції, як правило, це додатна дійсна константа в інтервалі ($\varphi \leq 2.0$).

3. Над мутантним вектором виконують операцію кросовера (схрещування). В ході неї деякі координати мутантного вектора заміщують відповідними координатами з базового вектора. Кожна координата заміщується з певною ймовірністю (ρ), яка також є параметром налаштування методу диференціальної еволюції.

Отриманий після схрещування вектор називається *пробним*. Якщо він виявляється кращим за базовий (значення цільової функції покращилося), то в новому поколінні базовий вектор замінюється на пробний, в іншому випадку він зберігається в новому поколінні.

4. На кожній епосі еволюційного процесу або із заданою періодичністю визначають кращий вектор покоління з метою контролю швидкості пошуку оптимального розв'язку. Умовами закінчення моделювання можуть бути наступні:

- вичерпана задана гранична кількість епох еволюції;
- вичерпано заданий граничний фізичний розрахунковий час;
- значення критерію оптимізації кращого вектора покоління не змінюється протягом заданої граничної кількості епох еволюції;
- досягнуто задовільне значення критерію оптимізації.

Вибір коефіцієнта φ і сталої кросовера ρ здійснюється емпірично, оскільки багато в чому залежить від рельєфу поверхні критерію оптимізації.

3. Генетичний алгоритм та його модифікації

Генетичні алгоритми – це процедури пошуку розв'язків, засновані на механізмах природного добору і успадкування. У них використовується еволюційний принцип виживання найбільш пристосованих особин. При цьому генетичним алгоритмам притаманні такі властивості, що характеризують їх стійкість:

- 1) кодування параметрів (обробляються не значення параметрів самої задачі, а їх закодована форма);
- 2) операції на популяціях (пошук розв'язку здійснюється виходячи не з єдиної точки, а з їх деякої популяції);
- 3) використання мінімуму інформації про завдання (використовується тільки цільова функція пошуку, а не її похідна або якась інша додаткова інформація);
- 4) рандомізація (застосовуються ймовірнісні, а не детерміновані правила вибору).

При описі генетичних алгоритмів застосовуються визначення, запозичені з генетики (популяція, хромосома і т.д.) – відповідні цим термінам визначення з технічного лексикона (ланцюг, двійкова послідовність і т.д.).

Популяцією є кінцева множина осіб, які в генетичних алгоритмах подають у вигляді хромосом із закодованими в них множинами параметрів задачі, тобто розв'язків, які інакше називаються точками в просторі пошуку. У деяких роботах особини називаються організмами.

Хромосоми (інші назви – ланцюжки або кодові послідовності) – це впорядковані послідовності генів. Ген (також називаний властивістю, знаком або детектором) – це атомарний елемент генотипу, зокрема, хромосоми. Під генотипом або структурою розуміється набір хромосом даної особини. Отже, особинами популяції можуть бути генотипи або одиничні хромосоми (в досить поширеному випадку, коли генотип складається з однієї хромосоми).

Фенотип – це набір значень, які відповідають даному генотипу, тобто декодована структура або множина параметрів задачі.

Алель – це значення конкретного гена, також визначається як значення властивості або варіант властивості. Локус чи позиція вказує місце розміщення даного гена в хромосомі (ланцюжку). Множина позицій генів в генетичних алгоритмах називається локами (рис. 15.1).

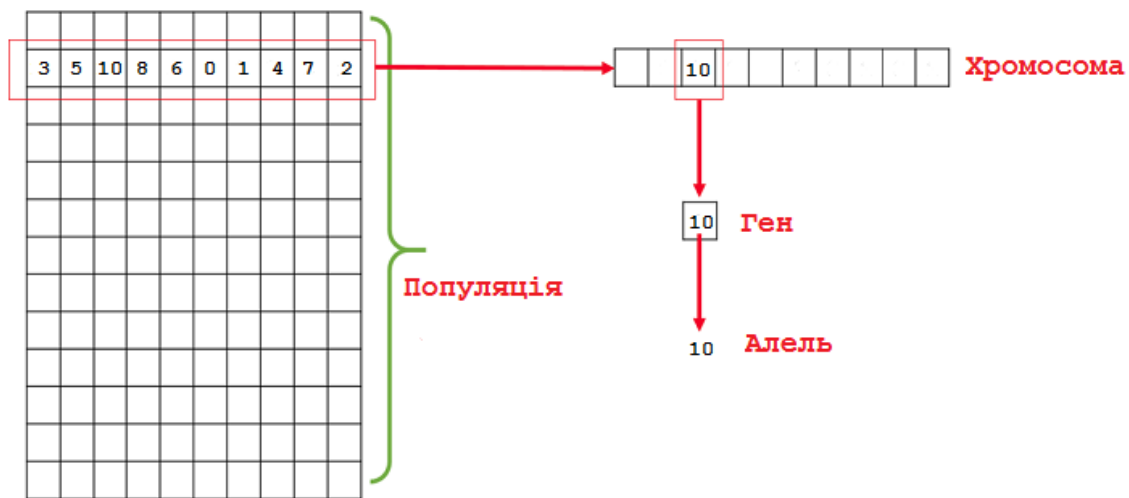


Рис. 15.1. Склад популяції

Дуже важливим поняттям в генетичних алгоритмах вважається функція пристосованості, або інакше – оцінки. Вона являє міру пристосованості даної особини в популяції і відіграє найважливішу роль, оскільки дозволяє оцінити ступінь пристосованості конкретних особин в популяції і вибрати з них найбільш пристосовані (тобто тих, що мають найбільші значення функції пристосованості) відповідно до еволюційного принципу виживання «найсильніших» (найкраще пристосованих). Функція пристосованості також отримала свою назву безпосередньо з генетики. Вона впливає на функціонування генетичних алгоритмів і повинна мати точне і коректне визначення. Наприклад, в завданнях оптимізації функція пристосованості, як правило, оптимізується (точніше кажучи, максимізується) і називається

цільовою. У задачах мінімізації цільова функція перетворюється, і проблема зводиться до максимізації. В теорії управління вона може приймати вигляд похибки, а в теорії ігор – вартісної функції. На кожній ітерації генетичного алгоритму пристосованість кожної особини даної популяції оцінюється за допомогою функції пристосованості, і на цій основі створюється наступна популяція особин, що складають множину потенційних розв'язків проблеми, наприклад, задачі оптимізації.

Чергова популяція в генетичному алгоритмі називається поколінням, а до новостворюваної популяції особин застосовується термін «нове покоління» або «покоління нащадків».

Основний класичний генетичний алгоритм (також званий елементарним або простим генетичним алгоритмом) складається з наступних кроків (див. рис. 15.2):

- 1) ініціалізація, або вибір вихідної популяції хромосом;
- 2) оцінка пристосованості хромосом в популяції;
- 3) перевірка умови зупинки алгоритму;
- 4) селекція хромосом;
- 5) застосування генетичних операторів;
- 6) формування нової популяції;
- 7) вибір «найкращої» хромосоми.

Ініціалізація, тобто формування вихідної популяції, полягає у випадковому виборі заданої кількості хромосом (особин), які подаються двійковими послідовностями фіксованої довжини.

Оцінювання пристосованості хромосом в популяції складається з розрахунку функції пристосованості для кожної хромосоми цієї популяції. Чим більше значення цієї функції, тим вищою є «якість» хромосоми. Форма функції пристосованості залежить від характеру розв'язуваної задачі. Передбачається, що функція пристосованості завжди набуває невід'ємних значень і для розв'язання оптимізаційної задачі потрібно її максимізувати. Якщо вихідна форма функції пристосованості не задовольняє цим умовам, то виконується відповідне перетворення (наприклад, задачу мінімізації функції можна легко звести до задачі максимізації).

Визначення умови зупинки генетичного алгоритму залежить від його конкретного застосування. В оптимізаційних задачах, якщо відомо максимальне (або мінімальне) значення функції пристосованості, то зупинка алгоритму може статися після досягнення очікуваного оптимального значення, можливо – із заданою точністю. Зупинка алгоритму також може відбуватися в разі, коли його виконання не веде до покращення вже досягнутого значення. Алгоритм може бути зупинений після закінчення певного часу виконання або після виконання заданої кількості ітерацій. Якщо умова зупинки виконана, то проводиться перехід до завершального етапу вибору «найкращої» хромосоми. В іншому випадку на наступному кроці виконується селекція.

Селекція хромосом полягає у виборі (за розрахованими на другому етапі значеннями функції пристосованості) тих хромосом, які будуть брати участь в створенні нащадків для наступної популяції, тобто для чергового покоління.

Такий вибір проводиться відповідно до принципу природного відбору, за яким найбільші шанси на участь в створенні нових особин мають хромосоми з найбільшими значеннями функції пристосованості.

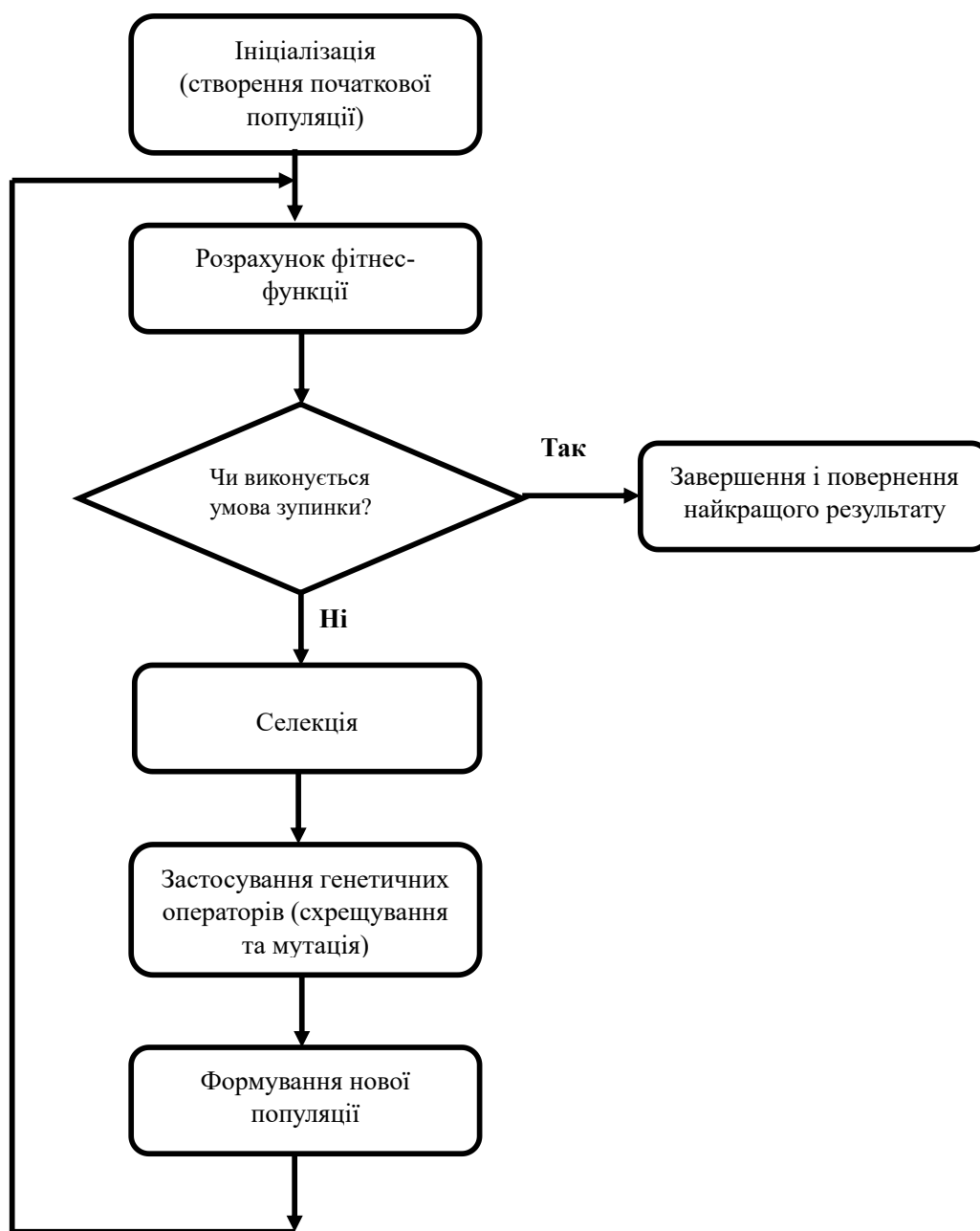


Рис. 15.2. Блок-схема генетичного алгоритму

Застосування генетичних операторів до хромосом, відібраних за допомогою селекції, спричиняє формування нової популяції нащадків на основі створеної на попередньому кроці батьківської популяції. У класичному генетичному алгоритмі застосовуються два основних генетичних оператори: схрещування і мутації. Однак перший відіграє другорядну роль в порівнянні з другим. Це означає, що схрещування в класичному генетичному алгоритмі проводиться практично завжди, тоді як мутація трапляється нечасто.

Імовірність схрещування, як правило, досить велика (зазвичай $0,5 \leq p_c \leq 1$), тоді як імовірність мутації встановлюється дуже малою (найчастіше $0 \leq p_m \leq 0,1$). Це впливає з аналогії зі світом живих організмів, де мутації відбуваються надзвичайно рідко. Імовірність p_m мутації може генеруватися, наприклад, випадковим вибором числа з інтервалу $[0, 1]$ для кожного гена і відбором для виконання цієї операції тих генів, для яких розігране число виявляється меншим або таким, що дорівнює значенню p_m .

Схрещування – це ключовий оператор генетичних алгоритмів, що визначає їх можливості і ефективність. Мутація грає більш обмежену роль. Вона вводить в популяцію деяку різноманітність і попереджає втрати, які могли б статися внаслідок виключення якого-небудь значимого гена в результаті схрещування. У генетичному алгоритмі мутація хромосом може виконуватися на популяції батьків перед схрещуванням або на популяції нащадків, утворених в результаті схрещування.

Для формування нової популяції хромосоми, отримані в результаті застосування генетичних операторів до хромосом тимчасової батьківської популяції, включаються до складу нової популяції. Вона стає так званою поточною популяцією для даної ітерації генетичного алгоритму. На кожній черговій ітерації розраховуються значення функції пристосованості для всіх хромосом цієї популяції, після чого перевіряється умова зупинки алгоритму і або фіксується результат у вигляді хромосоми з найбільшим значенням функції пристосованості, або здійснюється перехід до наступного кроку генетичного алгоритму, тобто до селекції. У класичному генетичному алгоритмі вся попередня популяція хромосом заміщується новою популяцією нащадків, що має ту ж чисельність.

На етапі вибору «найкращої» хромосоми перевіряється умова зупинки алгоритму і, якщо вона виконана, то слід вивести результат роботи, тобто надати шуканий розв'язок задачі. Кращим розв'язком вважається результат у вигляді хромосоми з найбільшим значенням функції пристосованості.

Оператори селекції

Найбільш популярним вважається так званий *метод рулетки*, який свою назву отримав за аналогією з відомою азартною грою. Кожна хромосома може бути зіставлена з сектором колеса рулетки, величина якого встановлюється пропорційно значенню функції пристосованості даної хромосоми. Тому, чим більшим є значення функції пристосованості, тим більший сектор на колесі рулетки. Все колесо рулетки відповідає сумі значень функції пристосованості всіх хромосом даної популяції. Кожній хромосомі, що позначається ch_i для $i = 1, 2, \dots, N$ (де N є чисельністю популяції), відповідає сектору колеса $v(ch_i)$, виражений у відсотках згідно з формулою (15.2):

$$v(ch_i) = \frac{F(ch_i)}{\sum_{i=1}^N F(ch_i)} 100\%, \quad (15.2)$$

де $F(ch_i)$ – значення функції пристосованості хромосоми ch_i , а число $p(ch_i) = \frac{F(ch_i)}{\sum_{i=1}^N F(ch_i)}$ називається ймовірністю селекції хромосоми ch_i . Таким чином, селекція хромосоми може бути подано як результат повороту колеса рулетки, оскільки «вигравша» (тобто обрана) хромосома відноситься до сектору, який випав на цьому колесі. Очевидно, що чим більший сектор, тим більша ймовірність «перемоги» відповідної хромосоми. Тому ймовірність вибору даної хромосоми виявляється пропорційною значенню її функції пристосованості. Якщо все коло колеса рулетки подати у вигляді цифрового інтервалу $[0, 100]$, то вибір хромосоми можна ототожнити з вибором числа з інтервалу $[a, b]$, де a і b позначають відповідно початок і закінчення фрагмента кола, відповідного цьому сектору колеса; очевидно, що $0 \leq a < b \leq 100$. У цьому випадку вибір за допомогою колеса рулетки зводиться до вибору числа з інтервалу $[0, 100]$, яке відповідає конкретній точці на колі колеса. В результаті процесу селекції створюється батьківська популяція, яку називають *батьківським пулом* з чисельністю N , що дорівнює чисельності поточної популяції.

Іншим популярним оператором селекції є ранжування. Попередній метод матиме проблеми, коли оцінки пристосованості сильно відрізняються. Наприклад, якщо пристосованість найкращої хромосоми становить 90 % від усієї рулетки, тоді інші хромосоми матимуть дуже мало шансів бути обраними. Оператор ранжування спочатку класифікує популяцію, а потім кожна хромосома отримує придатність за цим рейтингом. Найгірша матиме оцінку 1, друга за найгіршу 2 тощо, а найкращі матимуть оцінку N (кількість хромосом у популяції). Після цього всі хромосоми мають шанс бути відібраними. Але цей метод може призвести до повільнішої збіжності, оскільки найкращі хромосоми не так сильно відрізняються від інших. Оператор ранжування також працює з негативними значеннями оцінки пристосованості.

Оператори схрещування

За час існування генетичного алгоритму було запропоновано чимало підходів до схрещування. У всіх варіантах спільним є те, що на першому етапі схрещування вибираються пари хромосом з батьківської популяції (батьківського пулу). Як правило, батьки позначаються наступним чином:

$$\begin{aligned}x^{(1)} &= (x_1^{(1)}, \dots, x_n^{(1)}), \\x^{(2)} &= (x_1^{(2)}, \dots, x_n^{(2)}).\end{aligned}$$

Це тимчасова популяція, що складається з хромосом, відібраних в результаті селекції і призначених для подальших перетворень операторами схрещування і мутації з метою формування нової популяції нащадків. На даному етапі хромосоми з батьківської популяції випадковим чином об'єднуються в пари для утворення нащадків, які позначають так:

$$y^{(1)} = (y_1^{(1)}, \dots, y_n^{(1)}),$$

$$y^{(2)} = (y_1^{(2)}, \dots, y_n^{(2)}).$$

Далі для кожної пари відібраних у такий спосіб батьків застосовується один з варіантів кросовера.

Одноточковий кросовер. Розігрується позиція гена (локус) у хромосомі, що визначає так звану точку схрещування. Якщо хромосоми кожного з батьків складаються з L генів, то очевидно, що точка схрещування l_k є натуральним числом, меншим L . Тому фіксація точки схрещування зводиться до випадкового вибору числа з інтервалу $[1, L - 1]$. В результаті схрещування пари батьківських хромосом виходить така пара нащадків:

- 1) нащадок, хромосома якого на позиціях від 1 до l_k складається з генів першого з батьків, а на позиціях від $l_k + 1$ до L – з генів другого з батьків.
- 2) нащадок, хромосома якого на позиціях від 1 до l_k складається з генів другого батька, а на позиціях від $l_k + 1$ до L – з генів першого з батьків.

Дискретний кросовер. Схожий на одноточковий кросовер. В даному випадку також генерується комбінація батьківських генів. На відміну від одноточкового кросоверу, потомство складається лише з одного індивіду, а його кожний ген випадковим чином обирається серед батьківських з однаковою ймовірністю. Тобто нащадок може виглядати так:

$$y = (x_1^{(1)}, x_2^{(2)}, x_3^{(1)}, x_4^{(1)}, x_5^{(2)}, \dots, x_n^{(1)}).$$

Середній кросовер. Два батька $x^{(1)}$ та $x^{(2)}$ генерують нащадка у такий спосіб: $y_i = (x_i^{(1)} + x_i^{(2)})/2$ [56].

Лінійний кросовер. Дано два батька $x^{(1)}$ та $x^{(2)}$, нащадки-кандидати $y^{(1)}$, $y^{(2)}$ та $y^{(3)}$ мають форму:

$$\begin{aligned} y^{(1)} &= 0.5x^{(1)} + 0.5x^{(2)}, \\ y^{(2)} &= 1.5x^{(1)} - 0.5x^{(2)}, \\ y^{(3)} &= -0.5x^{(1)} + 1.5x^{(2)}. \end{aligned}$$

Лінійна комбінація батьків була виконана ген за геном. Двоє найкращих за пристосованістю серед $y^{(1)}$, $y^{(2)}$ та $y^{(3)}$ становлять потомство.

Цілий арифметичний кросовер. Це, мабуть, найбільш часто вживаний оператор, який працює, беручи зважену суму з однаковим $\alpha \in [0, 1]$ двох батьківських алелів для кожного гена.

$$\begin{aligned} y^{(1)} &= \alpha x^{(1)} + (1 - \alpha)x^{(2)}, \\ y^{(2)} &= \alpha x^{(2)} - (1 - \alpha)x^{(1)}. \end{aligned}$$

Локальний кросовер. Локальний кросовер такий самий, як і цілий арифметичний кросовер, за винятком того, що значення α вибрано випадковим чином для кожного розташування гена:

$$y_i^{(1)} = \alpha x_i^{(1)} + (1 - \alpha)x_i^{(2)},$$

$$y_i^{(2)} = \alpha x_i^{(2)} - (1 - \alpha)x_i^{(1)}.$$

Евристичний кросовер. Використовується для розв'язання нелінійних обмежених задач оптимізації. Також застосовується до умовних і безумовних задач оптимізації, що мають різний рівень складності.

На відміну від інших операторів кросинговеру, евристичний кросовер виробляє не більше одного нащадка від даної пари батьків, і він використовує значення функцій пристосованості батьків для отримання нащадків, які генеруються наступним чином:

$$y_i = \alpha (x_i^{(2)} - x_i^{(1)}) + x_i^{(2)},$$

тут α – рівномірно розподілене випадкове число в інтервалі $[0, 1]$.

Слід зауважити, що значення пристосованості батька $x^{(2)}$ не гірше, ніж у батька $x^{(1)}$. Якщо нащадки знаходяться за межами можливої області, генерується нове випадкове число для отримання іншого потомства з використанням:

$$y_i^{(2)} = x_i^{(2)} + \beta |x_i^{(1)} - x_i^{(2)}|,$$

тут β – випадкове число, яке генерується за допомогою функції розподілу Лапласа. Цей процес повторюється до k разів.

Оператор кросовера Лапласа. Цей орієнтований на батьків оператор кросовера генерує двох нащадків від пари батьків у такий спосіб. Спочатку генерується рівномірно розподілене випадкове число $\alpha \in [0, 1]$, далі – випадкове число β , яке отримують за розподілом Лапласа, просто інвертуючи функцію розподілу Лапласа таким чином:

$$\beta = \begin{cases} a - b \ln(\alpha), & \alpha \leq \frac{1}{2}, \\ a + b \ln(\alpha), & \alpha > \frac{1}{2}. \end{cases}$$

нащадки генеруються такими рівняннями:

$$y_i^{(1)} = x_i^{(1)} + \beta |x_i^{(1)} - x_i^{(2)}|,$$

$$y_i^{(2)} = x_i^{(2)} + \beta |x_i^{(1)} - x_i^{(2)}|.$$

Обидва нащадки розміщені симетрично щодо положення батьків. При менших значеннях b вони швидше за все будуть з'являтися поблизу батьків. Для більших значень b нащадки, як очікується, – далеко від батьків. Якщо батьки, за фіксованих значень a і b , знаходяться поруч один з одним, очікується, що й нащадки будуть поруч. В іншому випадку вони, імовірно, будуть далеко один від одного. Таким чином, запропонований оператор кросовера має самоадаптивну поведінку.

Оператор кросоверу BLX- α . Оператор кросовера BLX- α злучає двох батьків для отримання одного нащадка вигляду: $y = (y_1, \dots, y_i, \dots, y_n)$, де y_i – випадково (рівномірно) вибране число з інтервалу $[l_i - I\alpha, u_i + I\alpha]$, де

$$\begin{cases} u_i = \max(x_i^{(1)}, x_i^{(2)}), \\ l_i = \min(x_i^{(1)}, x_i^{(2)}), \\ I = u_i - l_i. \end{cases}$$

Оператор кросоверу PBX- α . Так само як в BLX- α , даний оператор злучає двох батьків для отримання одного нащадка того самого вигляду. Відмінність від BLX- α полягає в тому, що $y_i \in$ випадковим (рівномірно) вибраним числом з інтервалу $[l_i, u_i]$. Обмеження l_i й u_i задають таким чином:

$$\begin{cases} l_i = \max\{a_i, x_i^{(1)} - I\alpha\}, \\ u_i = \max\{b_i, x_i^{(2)} + I\alpha\}, \\ I = |x_i^{(1)} - x_i^{(2)}|. \end{cases}$$

Тут $x^{(1)}$ є матір'ю, а $x^{(2)}$, відповідно, батьком. Вони мають різні ролі, а їх взаємодія продемонстрована на рис.15.3. Матері вказують на місця пошуку, які отримують точки відбору. Батьки визначають масштаби цих територій. Коефіцієнт α використовується для визначення розповсюдження, пов'язаного з розподілом ймовірностей, який використовується для створення нащадків.



Рис. 15.3. Вплив PBX- α оператора

Оператори мутації.

Як вже було сказано, мутація створює в популяції новий генетичний матеріал для підтримки популяційного різноманіття. Це не що інше, як зміна випадкової частини ланцюга генів, що належить особині. Мутація відбувається з імовірністю мутації p_m . Мутаційний оператор забезпечує можливу мутацію деяких вибраних хромосом. Для мутації обрані лише випадкові хромосоми $p_m \times N$ у популяції.

Мутаційний оператор забезпечує можливу мутацію певних вибраних елементів деяких хромосом. Хромосома перед мутацією визначається як:

$$x = (x_1, \dots, x_i, \dots, x_n).$$

Елемент хромосоми, що мутується, позначається x_i . Хромосома після мутації визначається як:

$$x' = (x_1, \dots, x'_i, \dots, x_n).$$

Мутація Мекінена, Періо і Тойванена. Для розв'язання деяких мультидисциплінарних задач оптимізації форм в аеродинаміці та електромагнетиці був застосований новий оператор мутації.

З точки x мутована точка x' утворюється наступним чином. Нехай α рівномірно розподілене випадкове число, при якому $\alpha \in [0, 1]$. Тоді

$$x'_i = (1 - \hat{t})l_i + \hat{t}u_i,$$

де

$$\hat{t} = \begin{cases} t - t \left(\frac{t-r}{t} \right)^b, & \text{if } r < t, \\ t, & \text{if } r = t, \\ t + (1-t) \left(\frac{r-t}{1-t} \right)^b & \text{if } r > t. \end{cases}$$

Змінна t визначається як: $t = \frac{x-l_i}{u_i-l_i}$, тут l_i та u_i – це нижня та верхня межі i -ї змінної розв'язку відповідно. На відміну від нерівномірного мутаційного оператора, сила мутаційного оператора Мекінена, Періо та Тойванена не зменшується зі збільшенням кількості поколінь.

Нерівномірна мутація. Одним з найбільш використовуваних в дійсно-закодованих генетичних алгоритмах є оператор нерівномірної мутації, що призначена для тонкого налаштування можливостей, спрямованих на досягнення високої точності.

Елемент, що мутується у k -му поколінні, позначається x , $x_i \in [l_i, u_i]$. Як і раніше, l_i і u_i – це нижня та верхня межі x_i відповідно. Мутований сегмент задається наступним чином:

$$x'_i = \begin{cases} x_i + \Delta(k, u_i - x_i), & \text{if } \tau = 0, \\ x_i - \Delta(k, x_i - l_i), & \text{if } \tau = 1. \end{cases}$$

Тут τ є випадковим числом, яке набуває значення 0 або 1. Значення функції Δ обчислюється за формулою:

$$\Delta(k, y) = y \left(1 - \alpha \left(1 - \frac{k}{T} \right)^b \right).$$

Вище α – рівномірно розподілене випадкове число в інтервалі $[0, 1]$, T – максимальна кількість поколінь, b – параметр, що визначає ступінь неоднорідності. Ця операція повертає значення в діапазоні $[0, y]$, при якому ймовірність повернення числа, близького до нуля, зростає із збільшенням k . У початкових поколіннях неоднорідна мутація має тенденцію до рівномірного пошуку простору, а в наступних поколіннях – до локального.

Мутація потужності. Оператор мутації заснований на розподілі потужності. Для утворення мутованого індивіда використовується наступна формула:

$$x' = \begin{cases} x - s(x - l), & \text{if } t < \alpha, \\ x + s(u - x), & \text{if } t \geq \alpha, \end{cases}$$

де $t = \frac{x-l_i}{u_i-l_i}$, l і u – нижня і верхня межі змінної розв'язку, а α – рівномірно розподілене випадкове число між 0 і 1. Імовірність утворення мутованого розв'язку x' на лівій (правій) стороні x пропорційна відстані x від $l(u)$, крім того мутований індивід завжди допустимий.

Рівномірна мутація. Це одинарний оператор мутації, заснований на поданні з плаваючою точкою, який вибирає випадковий елемент $i \in (1, \dots, n)$ хромосоми і замінює його допустимим випадковим значенням (з рівномірного розподілу ймовірностей).

Оператор є важливим на ранніх етапах еволюційного процесу, оскільки розв'язкам дозволяється вільно переміщатися у просторі пошуку, і в тому випадку, коли початкова сукупність складається з декількох копій однієї і тієї ж точки. Це може трапитися в задачах оптимізації з обмеженнями, де користувачі вказують початкову точку процесу. Єдина відправна точка дозволяє розробити ітераційний процес, коли наступна ітерація починається з найкращої точки попередньої ітерації. Цей метод був використаний при розробці системи для роботи з нелінійними обмеженнями у просторах, які не обов'язково є опуклими.

Гауссова мутація. Схожа з оператором рівномірної мутації, який вибирає випадковий елемент $i \in (1, \dots, n)$ хромосоми і замінює його допустимим випадковим значенням. На відміну від попереднього, даний оператор вибирає випадкове значення з нормального розподілу ймовірностей.

Огляд гібридних модифікацій генетичного алгоритму

Гібридні генетичні алгоритми останнім часом викликають значний інтерес і все частіше використовуються для розв'язання реальних проблем.

Хоча генетичні алгоритми можуть швидко знаходити область, в якій існує глобальний оптимум, їм потрібно відносно багато часу, щоб знайти точний локальний оптимум в області збіжності. Поєднання генетичного алгоритму та локального методу пошуку може пришвидшити пошук точного глобального оптимуму. У такому гібриді застосування локального пошуку до рішень, які керуються генетичним алгоритмом в напрямку до найбільш перспективного регіону, може прискорити наближення до глобального оптимуму. Час, необхідний для досягнення глобального оптимуму, може бути додатково скорочений, якщо локальні методи пошуку та місцеві знання застосовуються для прискорення пошуку найбільш перспективного регіону на додачу до визначення глобального оптимуму, починаючи з його області привабливості. Існує кілька стратегій використання інформації про локальний пошук у гібриді, за допомогою якого звичайний генетичний пошук може бути доповненим.

Для використання локальної інформації було прийнято два основні підходи, засновані на моделях біологічного навчання. Перший отримав назву

підхід Ламарка⁴, другий –Болдвіна⁵, існує також третя модель, яка є сумішшою базових моделей.

Ламарківський підхід заснований на успадкуванні набутих характеристик, отриманих завдяки навчанню. Він змушує генетичну структуру відображати результат локального пошуку. Генетична структура особини та її придатність змінені відповідно до рішення, знайденого методом локального пошуку. У підході Ламарка локальний пошук використовується як генетичний оператор уточнення, який модифікує генетичну структуру особини і повертає її назад до генетичної популяції. Еволюція Ламарка може прискорити процес пошуку генетичного алгоритму. З іншого боку, при зміні генетичної структури особини можна порушити схему обробки, що погано впливатиме на досліджувальні здібності генетичних алгоритмів. Це може призвести до передчасної збіжності.

Навчання Болдвіна дозволяє покращити пристосованість особини за допомогою локального пошуку, тоді як генотип залишається незмінним. Таким чином, це покращує шанси розв'язку поширити свою структуру на наступні покоління. Як і природна еволюція, навчання не змінює генетичну структуру особини, проте збільшує шанси на її виживання. Підхід Болдвіна, на відміну від підходу Ламарка, не дозволяє батькам передавати свої вивчені або набуті характеристики своїм нащадкам. Натомість зберігається лише придатність після навчання. Локальний метод пошуку в підході Болдвіна зазвичай використовується як частина процесу оцінки особини. Локальний метод пошуку застосовує локальні знання для отримання нового показника придатності, який може бути використаний глобальним генетичним алгоритмом для оцінки здатності особини вдосконалюватися.

Гібридні моделі створюються з метою поєднання переваг обох форм моделей навчання. Поєднання двох підходів можна здійснити на різних рівнях. Гібридизація може бути застосована на індивідуальному рівні, де деякі особини еволюціонують, використовуючи підхід Ламарка, тоді як інші особини еволюціонують, відповідно до підходу Болдвіна. Інший рівень – це рівень генів, де низка генів еволюціонує за стратегією Ламарка, а решта згідно з підходом Болдвіна.

Іншим важливим аспектом гібридного підходу є встановлення балансу між розвідкою та експлуатацією, щоб мати змогу вирішувати глобальні проблеми оптимізації ефективно. Відповідно до гібридної теорії розв'язок бажаної якості може бути досягнутий одним із двох шляхів. Або метод глобального пошуку досягає розв'язку самостійно, або веде пошук до області привабливості, звідки вже локальний метод пошуку продовжує шлях до бажаного розв'язку.

⁴ Жан Батіст П'єр Антуан де Монé Ламарк (*Jean-Baptiste Pierre Antoine de Monet, Chevalier de Lamarck*) (1744 – 1829) – французький зоолог, анатом, натураліст. Одним з перших створив цілісну теорію еволюції живого світу

⁵ Джеймс Марк Болдвін (*James Mark Baldwin*) (1861 – 1934) – американський психолог, філософ, соціолог. Один із засновників психології особистості й соціальної психології в США

Окрім ролі генетичних операторів у систематичному дослідженні простору пошуку, вони також частково виконують роль локального пошуку з відносно низькою вартістю порівняно з більш точними методами локального пошуку. Неправильне використання дорогого для обчислення локального пошуку в гібриді може вилитися в марні витрати ресурсів. Алгоритм повинен допомагати розумно приймати рішення щодо використання обох методів, особливо коли обидва можуть досягти бажаного результату, беручи до уваги переваги та витрати на їх використання.

Розподіл часу гібриду між двома методами впливає на ефективність та результативність процесу пошуку, тому є важливим нюансом, з яким стикаються розробники гібридних генетичних алгоритмів. Одним з шляхів до встановлення такого балансу є контроль за частотою використання локального пошуку. Кількість неперервних поколінь, які генетичний алгоритм виконує перед застосуванням локального пошуку, зазвичай називають частотою локального пошуку. У традиційному гібридному генетичному алгоритмі частота локального пошуку дорівнює 1.

Взагалі, оптимальна частота локального пошуку залежить від функції і змінюється у процесі, оскільки оптимальний час, який слід витратити на локальні та глобальні алгоритми пошуку, залежить від розподілу особин в популяції. Так, в деяких наукових роботах використовувалася зміна коефіцієнта варіації функції придатності для визначення, чи досліджує генетичний алгоритм нові регіони простору пошуку або використовує вже відомі. На основі цієї інформації алгоритм віддавав перевагу виконанню генетичної або локальної ітерації. Такий алгоритм покладається на роль локального пошуку як інструменту покращення вибірки нових регіонів, які досліджуються у разі будь-якого збільшення цього коефіцієнта. Як тільки пошук розгалужується до локальної частини, для прийняття рішення про продовження локального або перехід до глобального пошуку використовується співвідношення витрат на поліпшення придатності як останніх генетичних, так і локальних ітерацій та максимальна кількість локальних ітерацій.

Іншим важливим аспектом встановлення якісного балансу є тривалість роботи локального пошуку. Вона впливає на баланс між глобальним дослідженням генетичним алгоритмом та локальним вдосконаленням в гібридних генетичних алгоритмах. На неперервних функціях локальний пошук зазвичай закінчується при досягненні локального оптимуму, коли довжина його кроку стає замалою.

Була проведена низка досліджень оптимальної тривалості локального пошуку і встановлено, що використання усіченої тривалості локального пошуку дає найкращі результати для функцій Гриванка, тоді як повний пошук дає кращі результати для функцій Растригіна. Дослідники дійшли висновку, що тривалість локального пошуку є важливим фактором, і гібридні генетичні алгоритми з повним локальним пошуком будуть найбільш ефективними при вирішенні нетривіальних проблем.

Висока вартість повного локального пошуку функцій має критично високий вплив на продуктивність алгоритму в цілому через будь-яке неналежне

використання локального пошуку. В такій ситуації постає питання про відбір особин, до яких буде застосовано локальний пошук. Так, у будь-якому гібридному алгоритмі локальний пошук може бути застосований як до кожної особини в популяції, так і лише до певної її частини.

У традиційних гібридних генетичних алгоритмах локальний пошук робиться для кожної особини в популяції. У цьому випадку локальний пошук може бути застосований до особин, які потрапляють в одну і ту саму область тяжіння пошукового простору, потрапляючи в той самий локальний оптимум. Можливість застосування локального пошуку для більш ніж однієї особини з однієї області може бути зменшена, якщо проводити локальний пошук лише невеликої частини популяції. Це також знижує шанси на непотрібний локальний пошук особин, які потрапляють у неперспективні регіони простору пошуку. Прийняття рішення про оптимальну частку популяції, яка повинна здійснювати локальний пошук, має вплив на кінцевий результат роботи гібриду.

Наприклад, для зменшення обчислювальних зусиль методу локального пошуку без будь-яких додаткових параметрів була запропонована нова схема гібридного генетичного алгоритму, яка називається “Гібридний генетичний алгоритм з найкращим потомством”, позначена *ВОНГА*, з новою індивідуальною процедурою навчання. *ВОНГА* виконує локальний пошук лише тоді, коли найкраще потомство (рішення) у популяції є також найкращим у поточній батьківській популяції. Коли з’являються такі найкращі нащадки, дуже ймовірно, що найліпше потомство знаходиться на новій, вищій горі або на новій нижній долині. Отже, процедура повністю узгоджується з початковою ідеєю гібридного генетичного алгоритму, яка пропонує спочатку дати можливість генетичному алгоритму дослідити широкий простір пошуку і, як тільки алгоритм знайде потенційне рішення пошуку, тоді локальний пошук проведе уточнення потенційного рішення.

Ще одним засобом для боротьби із зайвими обчислювальними витратами, пов’язаними з повторним застосуванням локального пошуку до особин, що потрапляють в одну і ту ж область тяжіння пошукового простору, було запропоновано нову модифікацію методу Ньютона (далі схема Ньютона). Алгоритм *HGDN* базується на двох основних компонентах: методі глобального пошуку, заснованому на генетичному алгоритмі, та методі локального пошуку. Для локального пошуку було використано схему Ньютона. Вона ефективно ідентифікує декілька локальних мінімумів або максимумів в безпосередній близькості від початкової точки. Ключем до успіху схеми Ньютона є те, що знайдені оптимуми «видаляються», тобто зсув розміщується там, де знаходився оптимум. Подальший пошук Ньютона не прямує до тієї самої точки, а знаходить інший оптимум або розбігається, тобто демонструє відсутність оптимумів поблизу особини.

Класичний метод Ньютона передбачає обчислення градієнта і гессіана у певній точці функції і використовує цю інформацію для прогнозування нового розташування особини шляхом розв’язування $H(x)\gamma = -\nabla f(x)$, де H – це

матриця других часткових похідних функції $H(x)_{ij} = \frac{\partial f(x)}{\partial x_i \partial x_j}$, x – поточна позиція, а γ – покращення від поточної до наступної позиції. Схема Ньютона працює наступним чином: за допомогою методу Ньютона шукаються стаціонарні точки функції f , тобто точки x^* , де $f'(x^*) = 0$. Нехай x_1, \dots, x_N – це стаціонарні точки, які вже були визначені, тоді подальші стаціонарні точки можуть бути знайдені, обчислюючи вектор спуску за допомогою наступної формули

$$\nabla f_{x_0}(x) = \frac{\nabla f(x)}{\prod_{i=1}^N \|x - x_i\|^2}. \quad (15.3)$$

Функція f за таким напрямком не матиме оптимумів у точках x_i , а, отже, локальний метод пошуку більше не буде прямувати до них. Принцип роботи такого підходу можна побачити на Рис. 15..

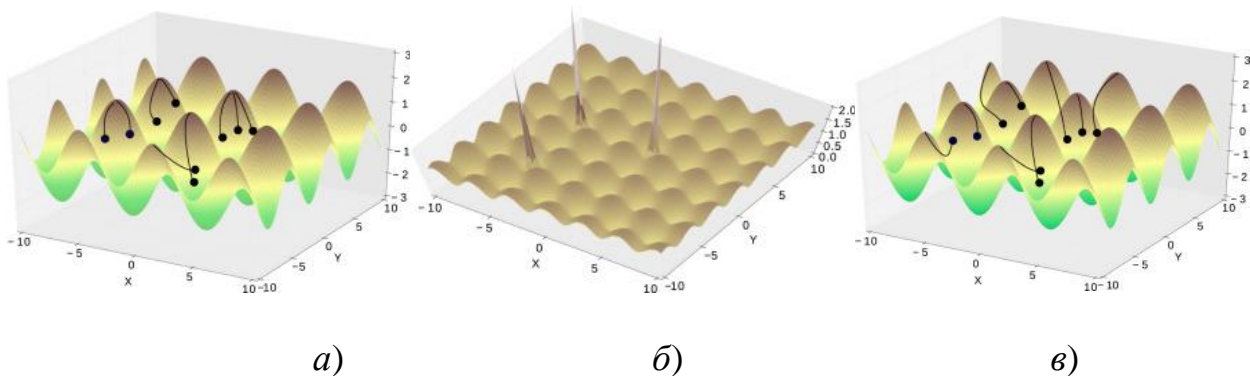


Рис. 15.4. Демонстрація роботи схеми Ньютона

Так, на рис. 15.4, *a* гібридний метод застосовує локальну оптимізацію до кожної особини. На наступній ітерації той самий оптимум може бути ідентифікований знову. Для уникнення подібної небажаної ситуації на рис. 15.4, *б* градієнт функції спущений в точках оптимуму функції, тобто корені видаляються і тому на рис. 15.4, *в* вже знайдені оптимуми неможливо ідентифікувати знову, а кожна особина може знайти нові оптимуми.

Щодо стратегії спуску запропонована вище формула (15.3) часто є не надійною для обчислень, тому розробляються альтернативні оператори. Більш загальною версією основного оператора спуску одного кореня є наступна:

$$\nabla f_{x_0}(x) = \frac{\nabla f(x)}{\|x - x_0\|^p}, \quad (15.4)$$

де $p \in \mathbb{N}$, але через її збіжність до 0 при $x \rightarrow \infty$, якщо f обмежена, було запропоновано введення зсуву оператора спуску в формі (15.5):

$$\nabla f_{x_0}(x) = \left(\frac{1}{\|x - x_0\|^p} + 1 \right) \nabla f(x). \quad (15.5)$$

Основним недоліком усіх вище наведених операторів є те, що вони змінюють функцію глобально, що в свою чергу призводить до виродження, якщо спуск застосовується багато разів. Це видно при застосуванні дефляції до двох різних коренів x_0 та x_1 :

$$\nabla f_{x_0, x_1}(x) = \left(\frac{1}{\|x - x_0\|^p} + 1 \right) \nabla f_{x_1}(x) = \left(\frac{1}{\|x - x_0\|^p} + 1 \right) \left(\frac{1}{\|x - x_1\|^p} + 1 \right) \nabla f(x)$$

Функція помножується на скалярний коефіцієнт більший за одиницю, який швидко зростає, якщо застосовується багато разів. Для розв'язання цієї проблеми було запропоновано локалізований оператор спуску, що використовує *bump*-функцію (плавна функція з компактним носієм), аби впливати лише на область, близьку до спуску. Нормована *bump*-функція в n -мірному просторі задається як (15.6):

$$b_{x_0}(x) = \begin{cases} \prod_{i=1}^n \frac{\exp\left(\frac{-\alpha}{r^2 - (x_i - x_{i0})^2}\right)}{\exp\left(\frac{-\alpha}{r^2}\right)}, & \text{if } x_{i0} - r < x_i < x_{i0} + r, \\ 0, & \text{else} \end{cases} \quad (15.6)$$

де x_0 – центр спуску, r – радіус спуску і α – коефіцієнт для регулювання форми *bump*-функції, як показано на рис. 15.5.

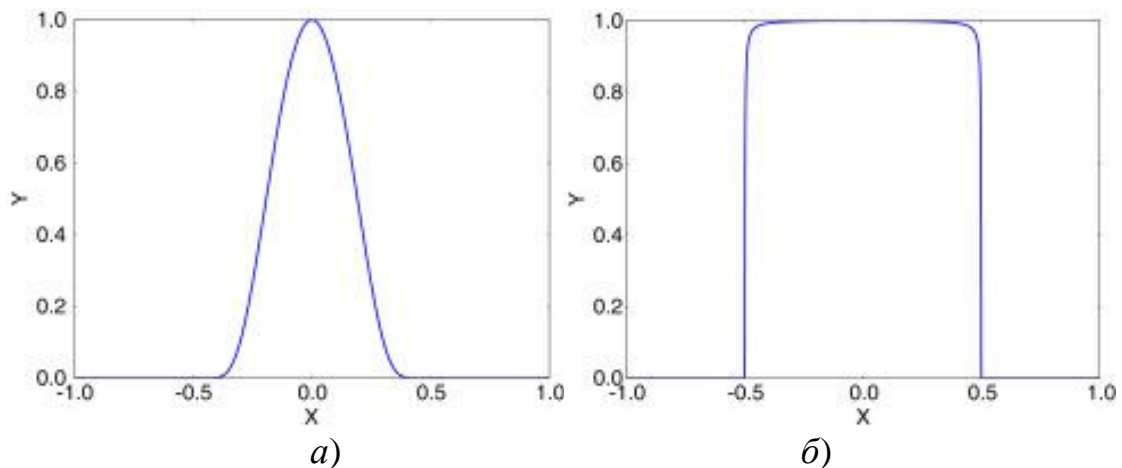


Рис.15.5. Приклад *bump*-функції а) $\alpha = 1$ б) $\alpha = 0.1$

Тоді функція спуску задається як $\nabla f_{x_0}(x) = \frac{\nabla f(x)}{1 - b_{x_0}(x)}$. Ефект різних операторів дефляції можна побачити на рис. 15.6.

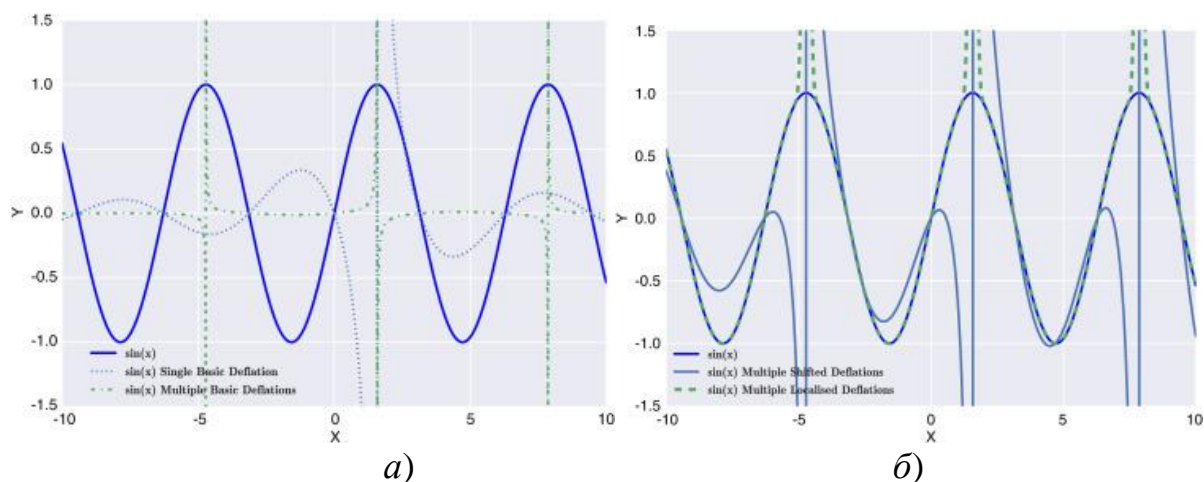


Рис. 15.6 – Вплив різних операторів спуску, застосованих до функції синуса

Так, на рис. 15.6, *a* маємо оригінальну синусову функцію та функцію, отриману з синусу за допомогою основного оператора спуску (15.3). Примітно те, що функція змінюється скрізь, коли застосовується операція спуску. Навіть, застосувавши спуск до функції синуса лише один раз, маємо значно менші амплітуди і наявність фазового зсуву, а при багаторазовому застосуванні спуску, функція синусу майже не відображає видимої амплітуди. Натомість на рис. 15.6, *б* використання неодноразово зміщеного оператора спуску (15.5) дає змінені амплітуди та фазовий зсув у сусідніх областях спущених точок в той час, як локалізований спуск (15.6), навпаки, залишає функцію незмінною поза певним радіусом.

Рекомендована література

1. Balamurugan R. Stellar-mass black hole optimization for biclustering microarray gene expression data / R. Balamurugan, A. M. Natarajan, K. Premalatha // *Applied Artificial Intelligence*. – 2015. – Vol. 29, No. 4. – P. 353–381.
2. Bianchi L. A survey on metaheuristics for stochastic combinatorial optimization / L. Bianchi, M. Dorigo, L. M. Gambardella, W. J. Gutjahr // *Natural Computing*. – 2009. – Vol. 8, No. 2. – P. 239 – 287.
3. Blum C. Metaheuristics in combinatorial optimization: overview and conceptual comparison / C. Blum, A. Roli // *ACM Comput. Surv.* – 2001. – Vol. 35. – P. 268 – 308.
4. *Handbook of metaheuristics* / ed. M. Gendreau, ed. J.-Y. Potvin. – New York : Springer, 2010. – 648 p.
5. Price K. *Differential evolution: a practical approach to global optimization* / K. Price, R. M. Storn, J. A. Lampinen. – Springer Science & Business Media, 2006. – 544 p.
6. Adewuya A. *New methods in genetic search with real-valued chromosomes* / A. Adewuya // 2005.
7. Dumitrescu D. *Evolutionary computation* / D. Dumitrescu, B. Lazzerini, L. C. Jain, [et al.]. – CRC Press, 2000.
8. García-Martínez C. Global and local real-coded genetic algorithms based on parent-centric crossover operators / C. García-Martínez, M. Lozano, F. Herrera, [et al.] // *European Journal of Operational Research*. – 2008. – Vol. 185, No. 3. – P. 1088 – 1113.

9. Arumugam M. S. New hybrid genetic operators for real coded genetic algorithm to compute optimal control of a class of hybrid systems / M. S. Arumugam, M. V. C. Rao, R. Palaniappan // *Applied Soft Computing*. – 2005. – Vol. 6, No. 1. – P. 38–52.
10. Kaelo P. Integrated crossover rules in real coded genetic algorithms / P. Kaelo, M. M. Ali // *European Journal of Operational Research*. – 2007. – Vol. 176, No. 1. – P. 60–76.
11. Deep K. A new mutation operator for real coded genetic algorithms / K. Deep, M. Thakur // *Applied Mathematics and Computation*. – 2007. – Vol. 193. – P. 211 – 230.
12. El-mihoub T. Hybrid genetic algorithms: a review / T. El-mihoub, A. Hopgood, L. Nolle, B. Alan // *Engineering Letters*. – 2006. – Vol. 3(2).
13. Booker L. Perspectives on adaptation in natural and artificial systems / L. Booker, S. Forrest, M. Mitchell, R. Riolo. – Oxford University Press, 2005. – 325 p.
14. Sung-Soon Choi A graph-based lamarckian-baldwinian hybrid for the sorting network problem / Sung-Soon Choi, Byung-Ro Moon // *IEEE Transactions on Evolutionary Computation*. – 2005. – Vol. 9, No. 1. – P. 105–114.
15. F. B. Espinoza, B. Minsker, and D. Goldberg A self adaptive hybrid genetic algorithm / F. B. Espinoza, B. Minsker, and D. Goldberg. – San Francisco, USA : Morgan Kaufmann Publishers, 2001.
16. Ishibuchi H. Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling / H. Ishibuchi, T. Yoshida, T. Murata // *IEEE Transactions on Evolutionary Computation*. – 2003. – Vol. 7, No. 2. – P. 204 – 223.
17. Wan W. An improved hybrid genetic algorithm with a new local search procedure / W. Wan, J. B. Birch. – Hindawi, 2013.
18. Moscato P. On evolution, search, optimization, genetic algorithms and martial arts - towards memetic algorithms / P. Moscato // *Caltech Concurrent Computation Program*. – 2000.
19. Noack M. M. Hybrid genetic deflated newton method for global optimisation / M. M. Noack, S. W. Funke // *Journal of Computational and Applied Mathematics*. – 2017. – Vol. 325. – P. 97 – 112.
20. Farrell P. E. Deflation techniques for finding distinct solutions of nonlinear partial differential equations / P. E. Farrell, Á. Birkisson, S. W. Funke // *SIAM Journal on Scientific Computing*. – 2015. – Vol. 37, No. 4. – P. A2026 – A2045.
21. Dai Y. H. A nonlinear conjugate gradient method with a strong global convergence property / Y. H. Dai, Y. Yuan // *SIAM Journal on Optimization*. – 1999. – Vol. 10, No. 1. – P. 177 –182.
22. Martens J. Deep learning via hessian-free optimization / J. Martens // P. 8.
23. Nash S. G. A survey of truncated-newton methods / S. G. Nash // *Journal of Computational and Applied Mathematics*. – 2000. – Vol. 124, No. 1. – P. 45 –59.
24. Kolda T. G. Optimization by direct search: new perspectives on some classical and modern methods / T. G. Kolda, R. M. Lewis, V. Torczon // *SIAM Review*. – 2003. – Vol. 45, No. 3. – P. 385 – 482.
25. Neumaier A. A comparison of complete global optimization solvers / A. Neumaier, O. Shcherbina, W. Huyer, T. Vinkó // *Mathematical Programming*. – 2005. – Vol. 103, No. 2. – P. 335 – 356.
26. Mongeau M. Comparison of public-domain software for black box global optimization / M. Mongeau, H. Karsenty, V. Rouzé, J. B. Hiriart-Urruty // *Optimization Methods and Software*. — 2000. – Vol. 13, No. 3. – P. 203 – 226.
27. Khompatraporn C. Comparative assessment of algorithms and software for global optimization / C. Khompatraporn, J. D. Pintér, Z. B. Zabinsky // *Journal of Global Optimization*. — 2005. – Vol. 31, No. 4. – P. 613 – 633.
28. Byrd R. H. A limited-memory algorithm for bound-constrained optimization / R. H. Byrd, L. Pei Huang, J. Nocedal. – Argonne National Laboratory, 1996.

29. Zhu C. Algorithm 778: l-bfgs-b: fortran subroutines for large-scale bound-constrained optimization / C. Zhu, R. H. Byrd, P. Lu, J. Nocedal // ACM Transactions on Mathematical Software. – 1997. – Vol. 23, No. 4. – P. 550–560.
30. Basak A. T. Dr Snehanshu Saha, Suryoday Psopy: a scipy compatible super fast python implementation for particle swarm optimization. / A. T. Basak Dr Snehanshu Saha, Suryoday.
31. Balas E. Guided local search with shifting bottleneck for job shop scheduling / E. Balas, A. Vazacopoulos // Management Science. – 1998. – Vol. 44, No. 2. – P. 262 – 275.
32. Beullens P. A guided local search heuristic for the capacitated arc routing problem / P. Beullens, L. Muyltermans, D. Cattrysse, D. Van Oudheusden // European Journal of Operational Research. – 2003. – Vol. 147, No. 3. – P. 629 – 643.
33. Faroe O. Guided local search for the three-dimensional bin-packing problem / O. Faroe, D. Pisinger, M. Zachariasen // INFORMS Journal on Computing. – 2003. – Vol. 15, No. 3. – P. 267 – 283.
34. Kilby P. Guided local search for the vehicle routing problem with time windows / P. Kilby, P. Prosser, P. Shaw // Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization / S. Voß, S. Martello, I. H. Osman, C. Roucairol. – Boston, MA : Springer US, 1999. – P. 473 – 486.
35. Vansteenwegen P. A guided local search metaheuristic for the team orienteering problem / P. Vansteenwegen, W. Souffriau, G. V. Berghe, D. V. Oudheusden // European Journal of Operational Research. – 2009. – Vol. 196, No. 1. – P. 118 –127.
36. Chris Voudouris and Edward Tsang Guided local search / Chris Voudouris and Edward Tsang. – Technical Report CSM, 1995.

Контрольні запитання

1. Що таке метаевристика?
2. Яка ідея методу імітації відпалу?
3. Що є основою методу рою частинок?
4. Сформулюйте основну ідею генетичних алгоритмів.
5. Які основні операції включає генетичний алгоритм розв'язання задачі оптимізації?
6. Що означає термін «пристосовність особини»?
7. В яких випадках доцільно використовувати гібридні евристичні алгоритми?
8. У чому полягає ідея гібридних алгоритмів?

ДОДАТОК А

Огляд тестових функцій

Тестові функції важливі для перевірки та порівняння ефективності алгоритмів оптимізації. У літературі наведено багато функцій для тестування. Існує багатий набір проблем безумовної оптимізації з різними властивостями з точки зору модальності, сепарабельності, масштабованості та ландшафту, що може бути справді корисним для неупередженого тестування алгоритмів.

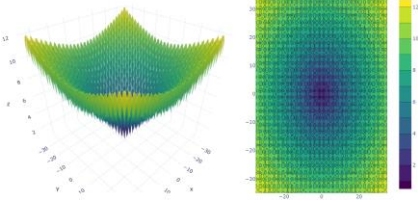
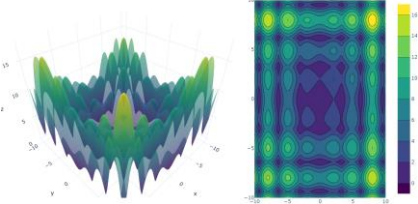
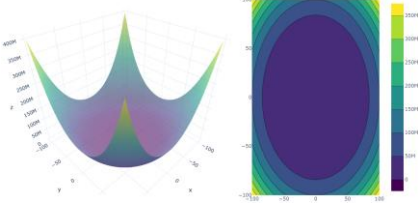
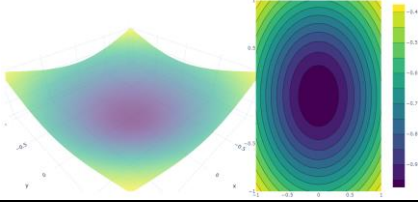
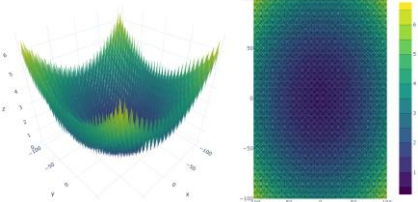
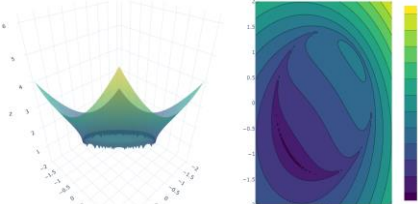
Функції з кількома локальними оптимумами називаються мультимодальними. Вони використовуються для перевірки здатності алгоритму виходити з будь-якого локального мінімуму. Якщо алгоритм має погано спроектований процес дослідження, тоді він не може ефективно досліджувати ландшафт функцій. Це, в свою чергу, призводить до того, що алгоритм застряє на локальному мінімумі. Мультимодальні функції з багатьма локальними мінімумами є одними з найскладніших класів задач для багатьох алгоритмів. Функції з рівними поверхнями також становлять труднощі, оскільки незмінність функції не дає алгоритму ніякої інформації для спрямування процесу пошуку до мінімумів (наприклад, Stepint або Powell Sum).

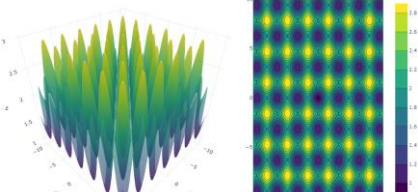
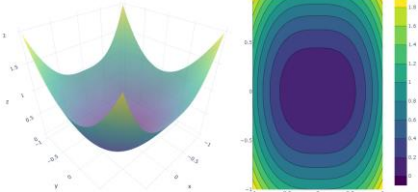
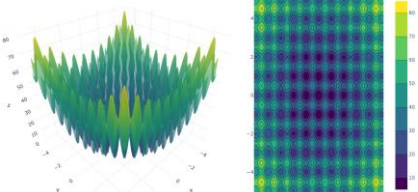
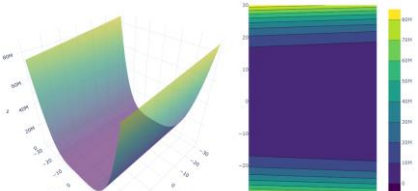
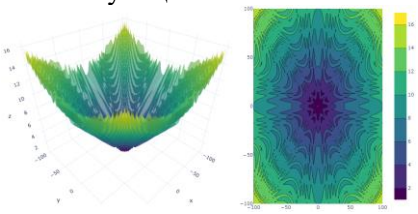
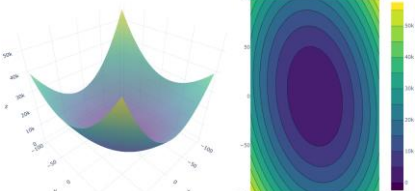
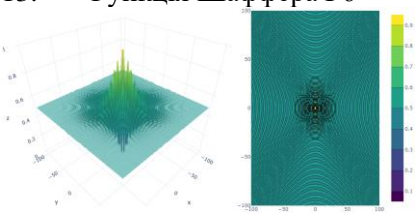
Інша група тестових задач ділиться на сепарабельні та несепарабельні. Загалом, для сепарабельних функцій вирішувати задачі оптимізації значно легше у порівнянні з несепарабельними, оскільки кожна змінна функції незалежна від інших. Якщо всі параметри або змінні незалежні, то може бути виконана послідовність n незалежних процесів оптимізації. Як результат, кожна змінна або параметр може бути оптимізовано незалежно один від одного.

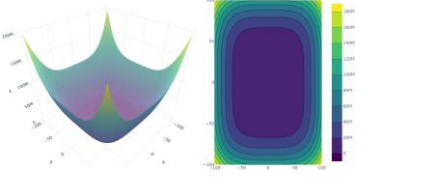
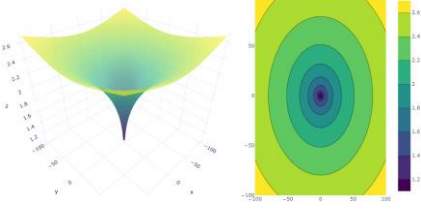
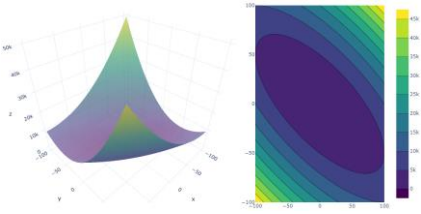
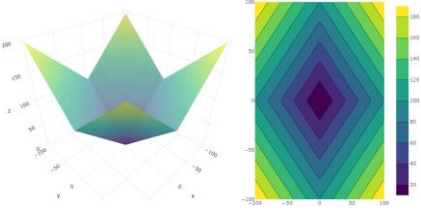
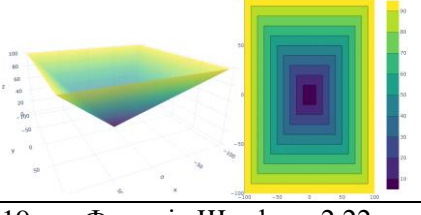


Розмірність простору пошуку є важливою проблемою. Із збільшенням кількості параметрів або розмірності простір пошуку також збільшується експоненційно. Для деяких функцій область, яка містить глобальні мінімуми, дуже мала, якщо порівнювати з цілим простором пошуку, наприклад, Powell. Для таких проблем, як Schaffer, глобальний мінімум знаходиться дуже близько до локальних мінімумів. Якщо алгоритм не може встигати за зміною напрямку у функціях з вузькою кривою долиною або не може ефективно досліджувати пошуковий простір, у випадку, якщо функції мають кілька глобальних мінімумів, він не зможе впоратися з такими проблемами.

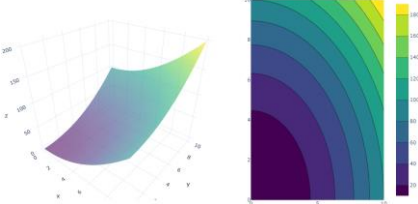
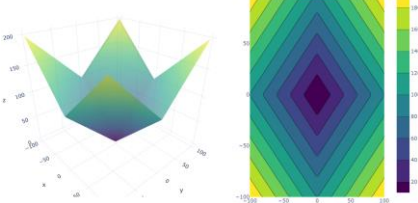
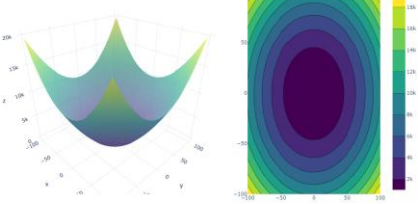
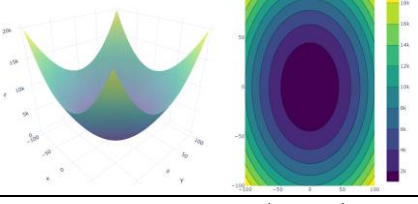
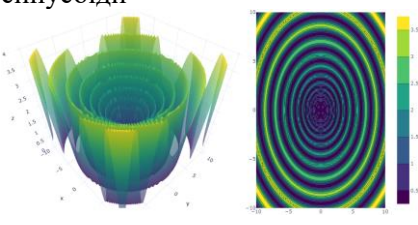
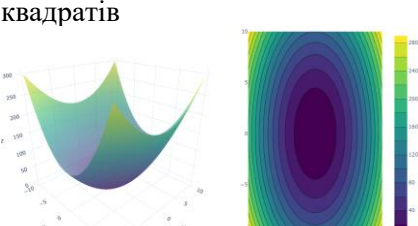
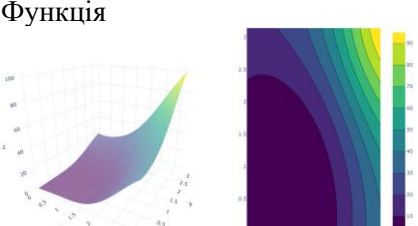
Ще однією проблемою алгоритмів є масштабування з різницею багатьох порядків між доменом та функцією гіперповерхні. Користуючись оглядом сучасних штучних ландшафтів, для перевірки ефективності алгоритмів були обрані 33 функції (табл. А.1). Розмірність проблеми позначимо D , а обмеження на розмір області проблеми та її оптимальний розв'язок $Lb \leq x_i \leq Ub$ та $f(x^*) = f(x_1^*, \dots, x_D^*)$ відповідно.

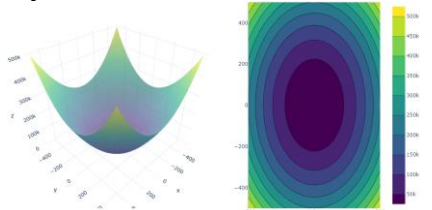
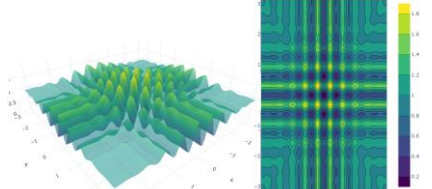
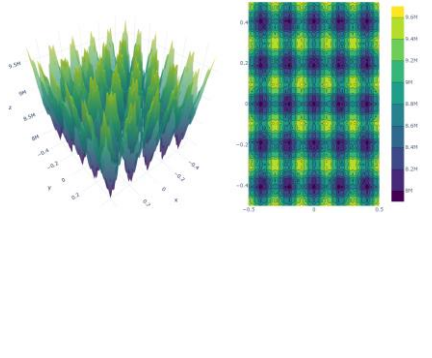
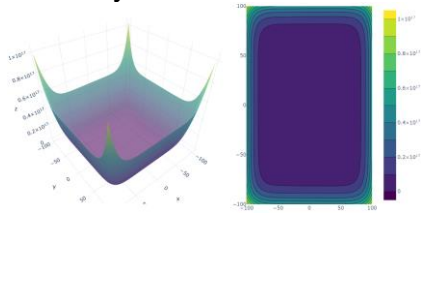
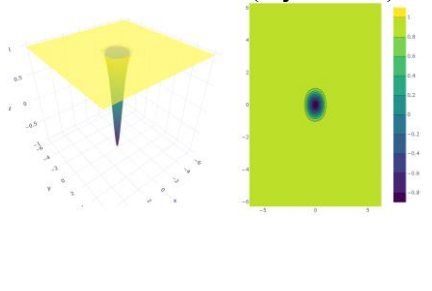
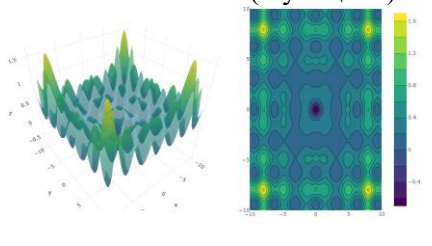
Штучні ландшафти

№, Назва, вигляд у 3Д та контурна мапа	Математична модель	Властивості
1. Функція Еклі 1 	$f(x) = -20e^{-0.02\sqrt{D^{-1}\sum_{i=1}^D x_i^2}} - e^{D^{-1}\sum_{i=1}^D \cos(2\pi x_i)} + 20 + e,$ $-35 \leq x_i \leq 35,$ $x^* = (0, \dots, 0), \quad f(x^*) = 0$	Неперервна, диференційована, несепарабельна, мультимодальна, невіпукла
2. Функція Alpine 1 	$f(x) = \sum_{i=1}^D x_i \sin(x_i) + 0.1x_i ,$ $-10 \leq x_i \leq 10,$ $x^* = (0, \dots, 0), f(x^*) = 0$	неперервна, недиференційована, несепарабельна, мультимодальна, невіпукла
3. Функція Чунга-Рейнольдса 	$f(x) = \left(\sum_{i=1}^D x_i^2\right)^2,$ $-100 \leq x_i \leq 100,$ $x^* = (0, \dots, 0), f(x^*) = 0$	Неперервна, диференційована, частково-сепарабельна, унімодальна, випукла
4. Експоненціальна функція 	$f(x) = -\exp\left(-0.5 \sum_{i=1}^D x_i^2\right),$ $-1 \leq x_i \leq 1,$ $x^* = (0, \dots, 0), f(x^*) = 1$	неперервна, диференційована, несепарабельна, унімодальна, випукла
5. Функція Griewank 	$f(x) = \sum_{i=1}^D \frac{x_i^2}{4000} - \prod \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1,$ $-100 \leq x_i \leq 100,$ $x^* = (0, \dots, 0), f(x^*) = 0$	неперервна, диференційована, несепарабельна, мультимодальна, невіпукла
6. Функція щасливого кота 	$f(x) = [(\ x\ ^2 - n)^2]^\alpha + \frac{1}{n} \left(\frac{1}{2} \ x\ ^2 + \sum_{i=1}^D x_i \right) + \frac{1}{2}$ $-2 \leq x_i \leq 2,$ $x^* = (-1, \dots, -1), f(x^*) = 0.$ <p>α – дійсний параметр</p>	неперервна, диференційована, несепарабельна, мультимодальна, невіпукла, параметрична

<p>7. Періодична функція</p> 	$f(x) = 1 + \sum_{i=1}^D \sin^2(x_i) - 0.1e^{(\sum_{i=1}^D x_i^2)}$ $-10 \leq x_i \leq 10,$ $x^* = (0, \dots, 0), f(x^*) = 0.9.$	<p>неперервна, диференційована, несепарабельна, мультимодальна, невишукла</p>
<p>8. Функція суми Пауелла</p> 	$f(x) = \sum_{i=1}^D x_i ^{i+1}$ $-1 \leq x_i \leq 1,$ $x^* = (0, \dots, 0), f(x^*) = 0.$	<p>неперервна, недиференційована, сепарабельна, унімодальна, вишукла</p>
<p>9. Функція Растрігіна</p> 	$f(x, y) = 10D + \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i))$ $-5.12 \leq x_i \leq 5.12,$ $x^* = (0, \dots, 0), f(x^*) = 0.$	<p>неперервна, диференційована, сепарабельна, мультимодальна, невишукла</p>
<p>10. Функція Розенброка</p> 	$f(x) = \sum_{i=1}^{D-1} \left[100(x_{i+1} - x_i^2)^2 \right. \\ \left. + (x_i - 1)^2 \right]$ $-30 \leq x_i \leq 30,$ $x^* = (1, \dots, 1), f(x^*) = 0.$	<p>неперервна, диференційована, несепарабельна, унімодальна, вишукла</p>
<p>11. Функція Саломона</p> 	$f(x) = 1 - \cos \left(2\pi \sqrt{\sum_{i=1}^D x_i^2} \right) \\ + 0.1 \sqrt{\sum_{i=1}^D x_i^2}$ $-100 \leq x_i \leq 100,$ $x^* = (0, \dots, 0), f(x^*) = 0.$	<p>неперервна, диференційована, несепарабельна, мультимодальна, невишукла</p>
<p>12. Функція Саргана</p> 	$f(x) = \sum_{i=1}^D \left(x_i^2 + 0.4 \sum_{j \neq i} x_i x_j \right)$ $-100 \leq x_i \leq 100,$ $x^* = (0, \dots, 0), f(x^*) = 0$	<p>неперервна, диференційована, несепарабельна, унімодальна, вишукла</p>
<p>13. Функція Шаффера F6</p> 	$f(x) = \sum_{i=1}^D 0.5 + \frac{\sin^2 \sqrt{x_i^2 + x_{i+1}^2} - 0.5}{[1 + 0.001(x_i^2 + x_{i+1}^2)]^2}$ $-100 \leq x_i \leq 100,$ $x^* = (0, \dots, 0), f(x^*) = 0$	<p>неперервна, диференційована, несепарабельна, мультимодальна, невишукла</p>

<p>14. Функція Шумера – Штайгліца</p> 	$f(x) = \sum_{i=1}^D x_i^4$ $x^* = (0, \dots, 0), f(x^*) = 0$	<p>неперервна, диференційована, сепарабельна, унімодальна, випукла</p>
<p>15. Функція Швевеля</p> 	$f(x) = \left(\sum_{i=1}^D x_i^2 \right)^\alpha$ $-100 \leq x_i \leq 100,$ $x^* = (0, \dots, 0), f(x^*) = 0$ $\alpha \geq 0 - \text{дійсний параметр}$	<p>неперервна, диференційована, частково-сепарабельна, унімодальна, випукла, параметрична</p>
<p>16. Функція Швевеля 1.2</p> 	$f(x) = \sum_{i=1}^D \left(\sum_{j=1}^i x_j \right)^2$ $-100 \leq x_i \leq 100,$ $x^* = (0, \dots, 0), f(x^*) = 0$	<p>неперервна, диференційована, несепарабельна, унімодальна, випукла</p>
<p>17. Функція Швевеля 2.20</p> 	$f(x) = - \sum_{i=1}^D x_i $ $-100 \leq x_i \leq 100,$ $x^* = (0, \dots, 0), f(x^*) = 0$	<p>неперервна, недиференційована, сепарабельна, унімодальна, випукла</p>
<p>18. Функція Швевеля 2.21</p> 	$f(x) = \max_{1 \leq i \leq D} x_i ,$ $-100 \leq x_i \leq 100,$ $x^* = (0, \dots, 0), f(x^*) = 0$	<p>неперервна, недиференційована, сепарабельна, унімодальна, випукла</p>
<p>19. Функція Швевеля 2.22</p> 	$f(x) = \sum_{i=1}^D x_i + \prod_{i=1}^D x_i $ $-100 \leq x_i \leq 100,$ $x^* = (0, \dots, 0), f(x^*) = 0$	<p>неперервна, недиференційована, несепарабельна, унімодальна, невипукла</p>
<p>20. Функція Швевеля 2.23</p> 	$f(x) = \sum_{i=1}^D x_i^4$ $-10 \leq x_i \leq 10,$ $x^* = (0, \dots, 0), f(x^*) = 0$	<p>неперервна, диференційована, сепарабельна, унімодальна, випукла</p>

<p>21. Функція сфери</p> 	$f(x) = \sum_{i=1}^D x_i^2$ $0 \leq x_i \leq 10,$ $x^* = (0, \dots, 0), f(x^*) = 0$	<p>неперервна, диференційована, сепарабельна, унімодальна, випукла</p>
<p>22. Крок Функція</p> 	$f(x) = \sum_{i=1}^D (x_i)$ $-100 \leq x_i \leq 100,$ $x^* = (0, \dots, 0), f(x^*) = 0$	<p>перервна, недиференційована, сепарабельна, унімодальна</p>
<p>23. Крок 2 Функція</p> 	$f(x) = \sum_{i=1}^D (x_i + 0.5)^2$ $-100 \leq x_i \leq 100,$ $x^* = (0.5, \dots, 0.5), f(x^*) = 0$	<p>перервна, недиференційована, сепарабельна, унімодальна</p>
<p>24. Крок 3 Функція</p> 	$f(x) = \sum_{i=1}^D (x_i^2)$ $-100 \leq x_i \leq 100,$ $x^* = (0, \dots, 0), f(x^*) = 0$	<p>перервна, недиференційована, сепарабельна, унімодальна</p>
<p>25. Розтягнута V-функція синусоїди</p> 	$f(x) = \sum_{i=1}^{D-1} (x_{i+1}^2 + x_i^2)^{0.25} \times$ $\left[\sin^2 \left\{ 50(x_{i+1}^2 + x_i^2)^{0.1} \right\} + 0.1 \right]$ $-10 \leq x_i \leq 10,$ $x^* = (0, \dots, 0), f(x^*) = 0$	<p>неперервна, диференційована, несепарабельна, мультимодальна, невипукла</p>
<p>26. Функція підсумованих квадратів</p> 	$f(x) = \sum_{i=1}^D ix_i^2$ $-10 \leq x_i \leq 10,$ $x^* = (0, \dots, 0), f(x^*) = 0$	<p>неперервна, диференційована, сепарабельна, унімодальна, випукла</p>
<p>27. Тригонометрична 1 Функція</p> 	$f(x) = \sum_{i=1}^D \left[D - \sum_{j=1}^D \cos x_j \right.$ $\left. + i(1 - \cos(x_i) - \sin(x_i)) \right]^2$ $0 \leq x_i \leq \pi,$ $x^* = (0, \dots, 0), f(x^*) = 0$	<p>неперервна, диференційована, несепарабельна, унімодальна, випукла</p>

<p>28. Тригонометрична 2 Функція</p> 	$f(x) = 1 + \sum_{i=1}^D 8 \sin^2[7(x_i - 0.9)^2]$ $+ 6 \sin^2[14(x_i - 0.9)^2] + (x_i - 0.9)^2$ $-500 \leq x_i \leq 500,$ $x^* = (0.9, \dots, 0.9), f(x^*) = 1$	<p>неперервна, диференційована, несепарабельна, мультимодальна, невишукла,</p>
<p>29. W / хвиляста функція</p> 	$f(x) = 1 - \frac{1}{D} \sum_{i=1}^D \cos(kx_i) e^{-\frac{x_i^2}{2}}$ $-\pi \leq x_i \leq \pi,$ $x^* = (0, \dots, 0), f(x^*) = 0$	<p>неперервна, диференційована, несепарабельна, масштабована, мультимодальна, невишукла</p>
<p>30. Функція Веерштраса</p> 	$f(x) = \sum_{i=1}^D \left[\sum_{k=0}^{kmax} a^k \times \right.$ $\left. \cos \left(2\pi b^k (x_i + 0.5) \right) \right.$ $\left. - D \sum_{k=0}^{kmax} a^k \cos \left(\pi b^k \right) \right]$ $-0.5 \leq x_i \leq 0.5,$ $x^* = (0, \dots, 0), f(x^*) = 0$	<p>неперервна, диференційована, сепарабельна, масштабована, мультимодальна, невишукла</p>
<p>31. Функція Вітлі</p> 	$f(x) =$ $\sum_{i=1}^D \sum_{j=1}^D \left[\frac{(100(x_i^2 - x_j)^2 + (1 - x_j)^2)^2}{4000} \right.$ $\left. - \cos \left(100(x_i^2 - x_j)^2 + (1 - x_j)^2 + 1 \right) \right]$ $x^* = (1, \dots, 1), f(x^*) = 0$	<p>неперервна, диференційована, несепарабельна, масштабована, мультимодальна, невишукла</p>
<p>32. Сінь-Ше Ян (Функція 3)</p> 	$f(x) = e^{-\sum_{i=1}^D \left(\frac{x_i}{\beta}\right)^{2m}} - 2e^{-\sum_{i=1}^D (x_i)^2}$ $\cdot \prod_{i=1}^D \cos^2(x_i)$ $-20 \leq x_i \leq 20,$ $x^* = (0, \dots, 0), f(x^*) = -1$ <p>$m = 5$ та $\beta = 15$ – параметри</p>	<p>неперервна, диференційована, несепарабельна, унімодальна, невишукла, параметрична</p>
<p>33. Сінь-Ше Ян (Функція 4)</p> 	$f(x) = \left[\sum_{i=1}^D \sin^2(x_i) - e^{-\sum_{i=1}^D x_i^2} \right] \times$ $e^{-\sum_{i=1}^D \sin^2 \sqrt{ x_i }}$ $-10 \leq x_i \leq 10,$ $x^* = (0, \dots, 0), f(x^*) = -1$	<p>неперервна, недиференційована, несепарабельна, масштабована, мультимодальна, невишукла</p>

ДОДАТОК Б

Додаткова література

1. Handbook of global optimization / ed. R. Horst, ed. P. Pardalos. – Springer US, 1995.
2. Horst R. Introduction to global optimization / R. Horst, P. Pardalos, N. V. Thoai. – Springer US, 2000.
3. Handbook of applied optimization / ed. P. M. Pardalos, ed. M. G. C. Resende. – Oxford, New York : Oxford University Press, 2002. – 116 p.
4. Pardalos P. M. Recent developments and trends in global optimization / P. M. Pardalos, H. E. Romeijn, H. Tuy // Journal of Computational and Applied Mathematics. – 2000. – Vol. 124, No. 1. – P. 209 – 228.
5. Spears W. M. An overview of evolutionary computation / W. M. Spears, K. A. De Jong, T. Bäck, [et al.]. – Berlin, Heidelberg : Springer, 1993.
6. Neri F. Recent advances in differential evolution: a survey and experimental analysis / F. Neri, V. Tirronen // 2010.
7. Siddique N. Nature inspired computing: an overview and some future directions / N. Siddique, H. Adeli // Cognitive Computation. – 2015. – Vol. 7, No. 6. – P. 706 – 714.
8. János D. Pintér, PhD, DSc Course on global optimization models, algorithms, software, and applications / János D. Pintér, PhD, DSc. – PCS Inc., Canada : 2014.
9. Zhigljavsky A. A. Theory of global random search / A. A. Zhigljavsky. – Springer Netherlands, 1991.
10. Boender C. G. E. Stochastic methods / C. G. E. Boender, H. E. Romeijn // Handbook of Global Optimization / R. Horst, P. M. Pardalos. – Boston, MA : Springer US, 1995. – P. 829 – 869.
11. Mockus J. Bayesian heuristic approach to global optimization and examples / J. Mockus // Journal of Global Optimization. – 2002. – Vol. 22, No. 1. – P. 191 – 203.
12. Diener I. Trajectory methods in global optimization / I. Diener // Handbook of Global Optimization / R. Horst, P. M. Pardalos. – Boston, MA : Springer US, 1995. – P. 649 – 668.
13. Forster, W. Homotopy methods / Forster, W. // Handbook of Global Optimization: Nonconvex Optimization and Its Applications. – Dordrecht, Netherlands : 1995. – P. 669 – 750.
14. Zheng Q. Integral global minimization: algorithms, implementations and numerical tests / Q. Zheng, D. Zhuang // Journal of Global Optimization. – 1995. – Vol. 7, No. 4. – P. 421 – 454.
15. Hichert J. Convergence speed of an integral method for computing the essential supremum / J. Hichert, A. Hoffmann, H. Phu. – 1997.
16. Lozano M. Real-coded memetic algorithms with crossover hill-climbing / M. Lozano, F. Herrera, N. Krasnogor, D. Molina // Evolutionary Computation. – 2004. – Vol. 12, No. 3. – P. 273 – 302.
17. Sudholt D. Local search in evolutionary algorithms: the impact of the local search frequency / D. Sudholt. – Berlin, Heidelberg : Springer, 2006.
18. Hedar A.-R. Hybrid simulated annealing and direct search method for nonlinear unconstrained global optimization / A.-R. Hedar, M. Fukushima // Optimization Methods and Software. – 2002. – Vol. 17, No. 5. – P. 891 – 912.
19. Hedar A.-R. Tabu search directed by direct search methods for nonlinear global optimization / A.-R. Hedar, M. Fukushima // European Journal of Operational Research. – 2006. – Vol. 170. – P. 329 – 349.
20. Hedar A.-R. Simplex coding genetic algorithm for the global optimization of nonlinear functions / A.-R. Hedar, M. Fukushima. – Berlin, Heidelberg : Springer, 2003.
21. Hedar A.-R. Derivative-free filter simulated annealing method for constrained continuous global optimization / A.-R. Hedar, M. Fukushima // Journal of Global Optimization. – 2006. – Vol. 35. – P. 521 – 549.
22. Jamil M. A literature survey of benchmark functions for global optimisation problems / M. Jamil, X. S. Yang // International Journal of Mathematical Modelling and Numerical Optimisation. – 2013. – Vol. 4, No. 2. – P. 150.

Навчальне видання

Коряшкіна Лариса Сергіївна
Ус Світлана Альбертівна

Практикум за курсом
«МЕТОДИ ОПТИМІЗАЦІЇ ТА ДОСЛІДЖЕННЯ ОПЕРАЦІЙ»
ЧАСТИНА II. НЕЛІНІЙНЕ ПРОГРАМУВАННЯ
Навчальний посібник

Редактор Є.М. Ільченко

Підп. до друку 2023. Формат 30x42/4.
Папір офсет. Ризографія. Ум. друк. арк. 12,8
Обл.-вид. арк. 16,7 Тираж пр. Зам. №

Підготовлено до друку та видруковано
в Національному технічному університеті
«Дніпровська політехніка»
Свідоцтво про внесення до Державного реєстру ДК № 1842.
49005, м. Дніпро, просп. Д. Яворницького, 19.

Навчальне видання

Коряшкіна Лариса Сергіївна
Ус Світлана Альбертівна

Практикум за курсом
«МЕТОДИ ОПТИМІЗАЦІЇ
ТА ДОСЛІДЖЕННЯ ОПЕРАЦІЙ»
ЧАСТИНА II. НЕЛІНІЙНЕ ПРОГРАМУВАННЯ

Навчальний посібник

Редактор Є.М. Ільченко

Підп. до друку 25.11.2023. Формат 60x90/8.
Папір офсет. Ризографія. Ум. друк. арк. 12,65
Обл.-вид. арк. 16,7 Тираж пр. Зам. №25/1123

Підготовлено до друку
в Національному технічному університеті
«Дніпровська політехніка»

Видавець «Свідлер А.Л.»

49041, м. Дніпро, а/с 2493, тел./факс +38 (067) 635-78-83
Свідоцтво про внесення до Державного реєстру суб'єктів видавничої
справи: Серія ДК № 3876 від 10.09.2010 р.

Надруковано в типографії видавництва «Свідлер А.Л.»
<http://www.garant-sv.com.ua>