МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Методи дослідження операцій

*Рекомендовано Методичною радою КПІ ім. Ігоря Сікорського
як навчальний посібник для студентів,
які навчаються за спеціальністю 122 «Комп'ютерні науки»*

Київ
КПІ ім. Ігоря Сікорського
2020

Методи дослідження операцій [Електронний ресурс]: навч. посіб. для студ. спеціальності 122 «Комп'ютерні науки» / КПІ ім. Ігоря Сікорського ; уклад.: В. О. Кузьміних, О. К. Молодід, Р. А. Тараненко. – Електронні текстові дані (1 файл: 2,185 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2020. – 117 с.

Гриф надано Методичною радою КПІ ім. Ігоря Сікорського (протокол № 2 від 1.10.2020 р.) за поданням Вченої ради ТЕФ (протокол № 11 від 25.06.2020 р.)

Електронне мережне навчальне видання

Методи дослідження операцій

Укладачі: *Кузьміних Валерій Олександрович*, канд. техн. наук, доц.
Молодід Олександр Кирилович
Тараненко Руслан Анатолійович

Відповідальний редактор: *Отрох С. І., д.т.н., доц. кафедри АПЕПС, ТЕФ*

Рецензенти: *Мельник Ю. В., д.т.н., с.н.с., директор Навчально-наукового інституту телекомунікацій Державного університету телекомунікацій (ННІТ ДУТ)*
Циганок В. В., д.т.н., с.н.с., завідувач лабораторії систем підтримки прийняття рішень відділу спеціалізованих засобів моделювання Інституту проблем реєстрації інформації (ІПРІ), НАН України

Невід'ємною складовою оволодіння кваліфікаційними навичками фахівців у сфері інформаційних технологій та систем є необхідність оволодіння засобами пошуку оптимальних рішень. Навчальний посібник «методи дослідження операцій» розглядає методи пошуку оптимальних рішень на основі математичного моделювання, статистичного моделювання та різних евристичних підходів. Матеріал викладено у обсязі необхідному в підготовці слухачів за спеціальністю 122 «Комп'ютерні науки». В навчальному посібнику подано загальний та практичний матеріал, подано табличні дані та гарно ілюстровано. Кожен розділ має питання для самоконтролю.

Зміст

Вступ 6

Розділ 1. Постановка задачі оптимізації 7

1.1. Математична постановка задач оптимізації. 7

1.2. Локальний та глобальний екстремум. 13

1.3. Збіжність методів оптимізації. 17

1.4. Умови зупинки розрахунку 18

1.5. Умови Оптимальності 19

Питання до розділу 1 22

Розділ 2. Оптимізація функції однієї змінної 23

2.1. Основні поняття 23

2.2. Характеристика методів 26

2.3. Методи виключення інтервалів 29

2.3.1. Метод рівномірного пошуку 29

2.3.2. Алгоритм ділення навпіл. 31

2.3.3. Метод Фібоначчі 33

2.3.4. Алгоритм золотого перетину 35

2.3.5. Порівняння ефективності алгоритмів одновимірної оптимізації 37

2.4. Методи поліноміальної апроксимації 39

2.4.1. Метод Пауелла (квадратичної апроксимації) 39

2.4.2. Метод кубічної апроксимації 40

2.5. Методи використання умов екстремуму 42

2.5.1. Метод хорд. 42

2.5.2. Пошук стаціонарної точки методом дотичних (Метод Ньютона) 44

2.5.3. Підвищення ефективності пошуку на основі умови Ліпшица 46

Питання до розділу 2 48

Розділ 3. Безумовна оптимізація функції кількох змінних 49

3.1. Основні поняття 49
3.1.1. Умови існування екстремуму 49
3.1.2. Рельєф функції 57
3.1.3. Характеристика методів 60
Питання до підрозділу 3.1. 61
3.2. Методи прямого пошуку 61
3.2.1. Метод пошуку по симплексу 61
3.2.2. Метод Нелдера–Міда 66
3.2.3. Спуск по координатах 69
3.2.4. Метод Хука-Дживса 73
3.2.5. Метод Розенброка 75
3.2.6. Метод спряжених напрямків 79
3.2.7. Метод паралельних дотичних 85
Питання до підрозділу 3.2. 87
3.3. Градієнтні методи 87
3.3.1. Метод найшвидшого спуску 87
3.3.2. Метод ярів 91
3.3.3. Метод Флетчера-Рівса 92
3.3.4. Метод Девідона-Флетчера-Пауелла 95
Питання до підрозділу 3.3. 96
3.4. Методи другого порядку 97
3.4.1. Метод Ньютона 97
3.4.2. Метод Левенберга-Марквардта 97
Питання до підрозділу 3.4. 99
3.5. Методи врахування обмежень 100
3.5.1. Зовнішні штрафні функції 100
3.5.2. Бар'єрні функції 101
Питання до підрозділу 3.5. 104
Розділ 4. Методи випадкового пошуку 104

4.1. Алгоритми локального пошуку 106
4.1.1. Простий випадковий пошук 106
4.1.2. Алгоритм парної проби 108
4.1.3. Алгоритм найкращої проби 109
4.1.4. Метод статичного градієнту 110
4.1.5. Алгоритм найкращої проби з направляючим гіперкубом 111
4.2. Алгоритми глобального пошуку 113
Питання до розділу 4 116
Перелік використаних джерел 116

ВСТУП

Математичні методи оптимізації та дослідження операцій – наука, яка вивчає оптимізацію функціонування складних систем. Теорія оптимізації - сукупність фундаментальних математичних результатів і чисельних методів, орієнтованих на пошук оптимального розв'язку практичних задач. На сьогодні теорія оптимізації широко використовується в усіх напрямках інженерної та економічної діяльності, наприклад, при проектуванні систем і їх складових частин, плануванні і аналізі роботи підприємств, управлінні динамічними системами [1; 2]. Важлива область застосування методів оптимізації пов'язана з удосконаленням існуючих систем виробництва і створенням ефективних виробничих планів для функціонування багатопрофільних техніко-економічних процесів.

У загальному випадку постановка задачі оптимізації полягає у наступному: знайти екстремум функції $f(x_1, x_2, ..., x_n)$ при наявності обмежень на її аргументи. Функція $f(x_1, x_2, ..., x_n)$ зветься цільовою функцією. Обмеження можуть мати вигляд рівностей ($h_k(x_1, x_2, ..., x_n) = 0$) або нерівностей ($g_m(x_1, x_2, ..., x_n) \geq 0$). Частинним випадком нерівностей є обмеження значень аргументів зверху та знизу: $x_{i,\min} \leq x_i \leq x_{i,\max}$. Розділ математики, присвячений розв'язуванню таких задач, зветься математичним програмуванням. Історично цей термін пов'язаний не зі складанням комп'ютерних програм, а з розробкою плану (програми) дій для отримання найкращого результату. Задачі оптимізації класифікуються в залежності від властивостей функцій f, h, g та розмірності вектору аргументів $x = (x_1, x_2, ..., x_n)$.

Якщо обмеження відсутні, то пошук екстремуму функції $f(x_1, x_2, ..., x_n)$ зветься задачею безумовної оптимізації, а інакше – умовної оптимізації. При

безумовній оптимізації окремо розглядають випадок цільової функції одного аргументу (n=1) та функції двох та більше аргументів (n≥2).

При наявності обмежень у залежності від характеру функцій f, h, g розрізняють задачі лінійного та нелінійного програмування. Якщо f, h, g є лінійними функціями, то відповідна задача зветься задачею лінійного програмування. Якщо хоча б одна з функцій f, h, g є нелінійною, задача зветься задачею нелінійного програмування.

Задачі нелінійного програмування в залежності від властивостей функцій f, h, g розділяються на більш вузькі класи задач, для кожного з яких розроблені свої числові методи.

Наприклад, якщо аргументи $x_1, x_2, \ldots, x_n$ приймають тільки цілі значення, задача зветься задачею цілочисельного програмування. В задачах дрібно лінійного програмування цільова функція є відношенням двох лінійних функцій, а функції h, g є лінійними. Якщо хоча б у одній з функцій f, h, g містяться випадкові величини, задача зветься задачею стохастичного програмування. Якщо цільова функція є квадратичною, а обмеження – лінійними, відповідна задача зветься задачею квадратичного програмування.

В даному навчальному посібнику висвітлені сучасні методи розв’язування задач одновимірної та багатовимірної, безумовної та умовної оптимізації [3; 4].

Посібник призначений для студентів напрямків підготовки бакалаврів ”Комп’ютерні науки” та ”Програмна інженерія”.

РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ ОПТИМІЗАЦІЇ

1.1. Математична постановка задач оптимізації

Оптимізація в широкому сенсі слова знаходить застосування в науці, техніці і в будь-якій області людської діяльності.

Оптимізація - цілеспрямована діяльність, яка полягає в отриманні найкращих результатів при відповідних умовах.

Пошуки оптимальних рішень привели до створення спеціальних математичних методів і вже в 18 столітті були закладені математичні основи оптимізації (варіаційне числення, чисельні методи та інші). Однак до другої половини 20 століття методи оптимізації в багатьох областях науки і техніки застосовувалися дуже рідко, оскільки практичне використання математичних методів оптимізації вимагало величезної обчислювальної роботи, яку без ЕОМ реалізувати було вкрай важко, а в ряді випадків – неможливо. Особливо великі труднощі виникали при вирішенні задач оптимізації процесів в авіації, електроніці, теплофізики [1; 4].

Слід виділити два основних мети завдань оптимізації:

– Пошук оптимальних значень параметрів і структури проектованого об'єкта (технічного, економічного) - задача синтезу об'єкта;

– Ідентифікація процесу з метою подальшого вивчення процесу або явища.

При постановці завдання оптимізації необхідно:

1. Наявність об'єкта оптимізації і мети оптимізації.

2. Наявність ресурсів оптимізації, під якими розуміють можливість вибору значень деяких параметрів оптимізуємого об'єкта. Об'єкт повинен мати певні ступені свободи.

3. Можливість кількісної оцінки оптимізується величини, оскільки тільки в цьому випадку можна порівнювати ефекти від вибору тих чи інших дій, що управляють.

4. Облік обмежень.

Незважаючи на те що прикладні завдання ставляться до абсолютно різних областей, вони мають загальну форму. Всі ці завдання можна класифікувати як завдання мінімізації речової функції f(x) N-мірного векторного аргументу $x = (x_1, x_2, \ldots, x_n)$, компоненти якого задовольняють

системі рівнянь $h_k(\text{x}) = 0$, набору нерівностей $g_i(\text{x}) < 0$, а також обмежені зверху і знизу, тобто $x_i^{min} \leq x_i \leq x_i^{max}$.

В подальшому викладі функцію f (x) будемо називати цільової функцією, рівняння $h_k(\text{x}) = 0$ - обмеженнями у вигляді рівностей, а нерівності $g_i(\text{x}) < 0$ – обмеженнями у вигляді нерівностей. При цьому передбачається, що всі фігурують в завданні функції є дійсними, а число обмежень кінцеве.

Задача загального вигляду:

Мінімізувати f (x) при обмеженнях

$h_k(\text{x}) = 0$, $k = 1, ..., K$,

$g_j(\text{x}) \geq 0$, $j = 1, ..., J$,

$x_i^{min} \leq x_i \leq x_i^{max}$, $i = 1, ..., N$

називається задачею оптимізації з обмеженнями або задачею умовної оптимізації. Задача, в якій немає обмежень, тобто

$x_i^{max} - x_i^{min} = \infty$, i = 1, ..., N,

називається оптимізаційною задачею без обмежень або задачею безумовної оптимізації.

Критерієм оптимальності називається кількісна оцінка якості об'єкта, що оптимізується.

На підставі обраного критерію оптимальності складається цільова функція, що представляє собою залежність критерію оптимальності від параметрів, що впливають на її значення.

Вид цільової функції визначається конкретним завданням оптимізації [4; 5].

Таким чином, завдання оптимізації зводиться до знаходження екстремуму цільової функції. Найбільш загальною постановкою оптимальної завдання є вираз критерію оптимальності у вигляді економічної оцінки (продуктивність, собівартість продукції, прибуток, рентабельність).

Вимоги до критерію оптимальності:

– Критерій оптимальності повинен виражатися кількісно.

– Критерій оптимальності повинен бути єдиним.

– Критерій оптимальності повинен відображати найбільш істотні сторони процесу.

Бажано щоб критерій оптимальності мав ясний фізичний зміст і легко розраховувався.

Будь-який об'єкт, що оптимізується, схематично можна уявити як показано на малюнку.

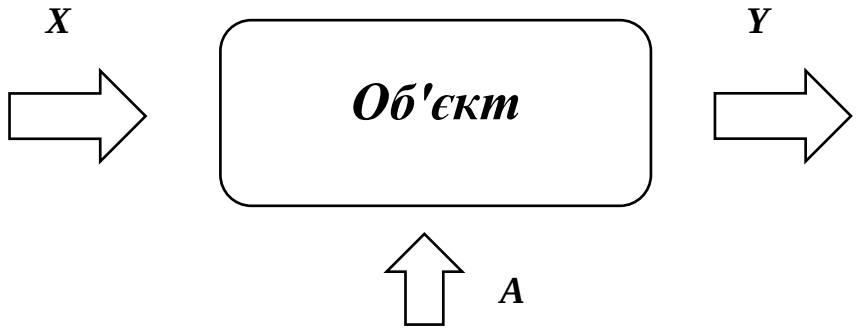


Рис. 1.1.1. Об'єкт, що оптимізується

X – вхідні параметри об'єкта;

Y – вихідні параметри об'єкта;

A – параметри оптимізації.

При постановці конкретних задач оптимізації критерій оптимальності повинен бути записаний у вигляді аналітичного виразу. У тому випадку, коли випадкові обурення невеликі і їх вплив на об'єкт можна не враховувати, критерій оптимальності може бути представлений як функція вхідних, вихідних і керуючих параметрів:

$R = R(X_1, X_2, \dots, X_N, Y_1, Y_2, \dots, Y_N, A_1, A_2, \dots, A_N)$;

Так $Y = f(A)$ то при фіксованих X можна записати:

$R = R(A)$

При цьому будь-яка зміна значень керуючих параметрів двояко позначається на величині R:

– прямо, бо керуючі параметри безпосередньо входять у вираз критерію оптимізації;

– побічно – через зміну вихідних параметрів процесу, які залежать від керуючих.

Якщо ж випадкові обурення досить великі і їх необхідно враховувати, то слід застосовувати експериментально - статистичні методи, які дозволять отримати модель об'єкта у вигляді функції

$$Y = f(X, A)$$

яка справедлива тільки для вивченої локальної області.

Тоді критерій оптимальності прийме наступний вигляд:

$$\mathrm{R} = \mathrm{R}(\mathrm{X}, \mathrm{A})$$

Для оптимізації замість математичної моделі можна використовувати і сам об'єкт, однак оптимізація досвідченим шляхом має ряд істотних недоліків:

1) необхідний реальний об'єкт;

2) необхідно змінювати технологічний режим в значних межах, що не завжди можливо;

3) тривалість випробувань і складність обробки даних. Наявність математичної моделі (за умови, що вона досить надійно описує процес) дозволяє значно простіше вирішити завдання оптимізації аналітичним або чисельним методами.

У завданнях оптимізації розрізняють прості і складні критерії оптимізації:

– критерій оптимальності називається простим, якщо потрібно визначити екстремум цільової функції без завдання умов на будь-які інші величини.

– критерій оптимальності називається складним, якщо вони являють собою сукупність декількох критеріїв.

Для вирішення завдання оптимізації необхідно:

1. Скласти математичну модель об'єкта оптимізації:

Y = f (X, A).

2. Вибрати критерій оптимальності і скласти цільову функцію:

R = f (X, A).

3. Встановити можливі обмеження, які повинні накладатися на змінні:

$\psi = f(X, A) = 0;$

$\varphi(X, A) < 0.$

4. Вибрати метод оптимізації, який дозволить знайти екстремальні значення шуканих величин.

Прийнято розрізняти завдання статичної оптимізації для процесів, що протікають в сталих режимах, і завдання динамічної оптимізації.

У першому випадку вирішуються питання створення і реалізації оптимальної моделі процесу, у другому - завдання створення і реалізації системи оптимального управління процесом при невстановлених режимах експлуатації [5].

Завдання оптимізації діляться на такі типи:

– Структурна і параметрична – по області зміни.

– Цілочисельна і дійсна (безперервні) – за типом параметрів.

– Аналітична, регулярна (детермінована) і стохастична – за методом пошуку результатів.

– Умовна і безумовна – за обмеженнями на область допустимих значень.

– Мінімізації і максимізації – за видом цільової функції.

– Локальна і глобальна по – за розміром області допустимих значень.

– Одновимірна і багатовимірні - за кількістю оптимізуються параметрів.

– Лінійний і нелінійні - по виду цільової функції (функції якості) і обмежень.

– Та інші.

Серед методів оптимізації можна виділити такі важливі групи:

– Аналітичні та пошукові.

– Пошукові: регулярні та стохастичні.

– Методи регулярні: нульового, першого порядку та другого порядку.

– Стохастичні: лінійний та адаптивні.

1.2. Локальний та глобальний екстремум

Сама по собі постановка задачі оптимізації проста і природна: задані безліч і функція, потрібно знайти точки мінімуму або максимуму.

Домовимося записувати завдання на мінімум у вигляді:

$$f(x) \to \min_{x \in X} \qquad (1.2.1)$$

де $f(x)$ - цільова функція;

X - допустима множина;

$\forall x \in R$ - допустима точка задачі.

Квантор загальності (позначення: ∀) - це умова, вірна для всіх позначених елементів;

Квантор існування (позначення: ∃) - одномісний предикат визначає відношення приналежності деякій множині.

В основному ми будемо мати справу з кінцевомірними завданнями оптимізації, тобто до завдань, в яких допустима безліч лежить в евклідовому просторі R^n ($x \in R^n$).

Точка $x^* \in X$, що є рішенням задачі (1.2.1), може бути точкою глобального або локального мінімуму.

Визначення.

Точка $x^* \in X$ називається:

1) точкою глобального мінімуму функції на множині X або глобальним рішенням задачі, якщо

$f(x^*) \leq f(x)$ при $\forall x \in X$. *(1.2.2)*

2) точкою локального мінімуму функції $f(x)$ на множині X або локальним рішенням задачі, якщо $\exists \varepsilon > 0$, таке що

для $\forall x \in X \cap U_\varepsilon(x^*)$, $f(x^*) \leq f(x)$, *(1.2.3)*

де $U_\varepsilon(x^*) = \{x \in R^n | \|x - x^*\| \leq \varepsilon\}$ – шар радиусу $\varepsilon > 0$ з центром в x^*.

Якщо нерівність в *(1.2.2)* або в *(1.2.3)* виконується як сувора при $x \neq x^*$, то x^* це точка суваорого мінімуму (суворе рішення) в глобальному або локальному розумінні.

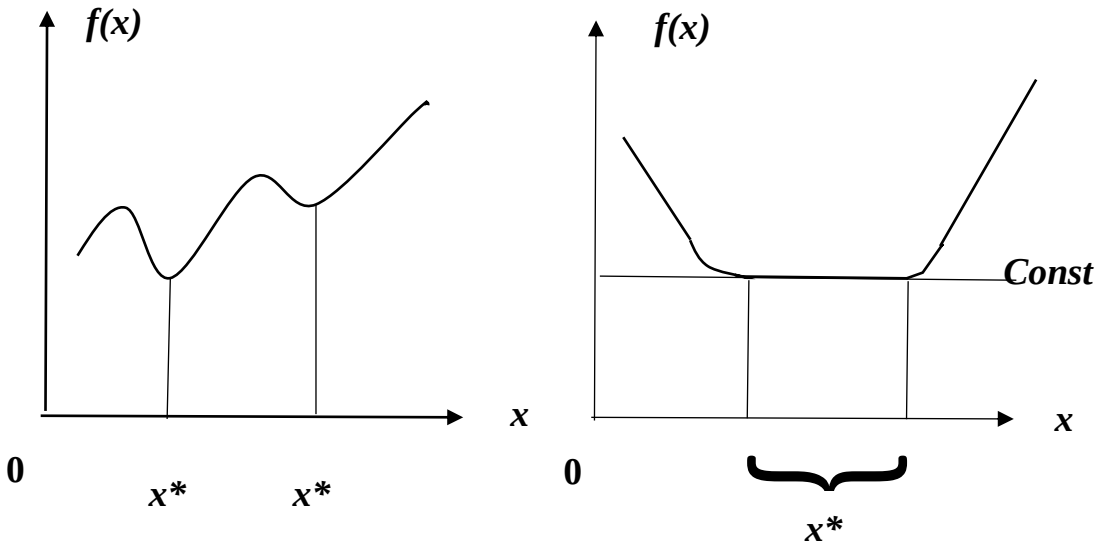


Рис. 1.2.1. Види мінімуму

Ясно, що глобальне рішення є і локальним; зворотне невірно.

Для відображення того факту, що точка $x^* \in X$ є точкою глобального мінімуму функції зазвичай використовується запис $f(x^*) = \min\limits_{x \in X} f(x)$ або еквівалентний їй запис

$x^* = arg \min\limits_{x \in X} f(x)$.

Точки мінімуму і максимуму функції на множині, називаються також точками екстремуму, а самі задачі - екстремальними задачами.

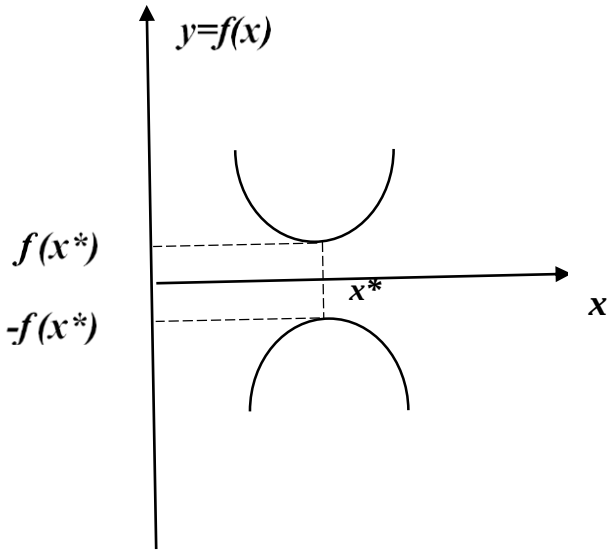


Рис. 1.2.2. Мінімум та максимум.

При вивченні задач оптимізації в першу чергу виникає питання про існування рішення. Відповідь на це питання дає теорема Вейерштрасса. Яка може бути сформулювала наступним чином:

Нехай X - компакт в R^n, $f(x)$ - безперервна функція на множині X. Тоді точка глобального мінімуму функції $f(x)$ на X існує *(глобальне рішення задачі існує)*.

У більшості випадків завдання оптимізації не вдається вирішити спираючись на необхідні і достатні умови оптимальності або на геометричну інтерпретацію задачі, і доводиться її вирішувати чисельно із застосуванням обчислювальної техніки. Причому, найбільш ефективними виявляються методи, розроблені спеціально для вирішення конкретного класу задач оптимізації, так як вони дозволяють повніше врахувати її специфіку [6].

Будь-який чисельний метод має два етапи:

– перший етап будь-якого чисельного методу (алгоритму) рішення задачі оптимізації заснований на точному або наближеному обчисленні її характеристик (значень цільової функції; значень функцій, які задають допустиму безліч, а також їх похідних);

– у другому етапі, на підставі отриманої інформації будується наближення до вирішення завдання - шуканої точцки мінімуму x^*, або, якщо така точка не єдина, до безлічі точок мінімуму.

Іноді, якщо тільки це потрібно, будується і наближення до мінімального значення цільової функції $f(x^*) = \min\limits_{x \in X} f(x)$.

Для кожної конкретної задачі питання про те, які характеристики слід вибрати для обчислення, вирішується в залежності від властивостей функції, що мінімізується, обмежень і наявних можливостей по зберіганню і обробці інформації.

Залежно від того які характеристики, зокрема, цільової функції беруться, алгоритми поділяються на алгоритми:

– нульового порядку - в них використовується інформація тільки про значення функції, що мінімізується;

– першого порядку - використовують інформацію також і про значення перших похідних;

– другого порядку - використовують, крім того, інформацію про других похідних;

і так далі.

Коли вирішено питання про те, які саме характеристики розв'язуваної задачі слід обчислювати, то для завдання алгоритму досить вказати спосіб вибору точок обчислення.

Залежно від способу вибору точок обчислення, алгоритми поділяються на пасивні і активні (послідовні).

У пасивних алгоритмах всі крапки вибираються одночасно до початку обчислень.

В активних (послідовних) алгоритмах точки обчислення x^{i+1} вибираються по черзі, тобто точка вибирається, коли вже обрані точки попередніх обчислень. $x^1, x^2, \dots, x^i$.

Надалі для знаходження x^{k+1} будемо користуватися співвідношенням виду $x^{k+1} = x^k + a_k * h^k, \ \ a_k \in R, k = 0,1,2, \ldots$ *(1.2.4)*

При цьому конкретний алгоритм визначається:

– заданням точки x^0;

– правилами вибору векторів h^k і чисел a_k на основі отриманої в результаті обчислень інформації.

Вектор h^k визначає напрямок $k + 1$-го кроку методу мінімізації, а коефіцієнт a_k - довжину цього кроку. Зазвичай назва методу мінімізації визначається способом вибору, а його різні варіанти зв'язуються з різними способами вибору.

Поряд з терміном крок методу ми будемо користуватися також терміном ітерація методу.

Серед методів мінімізації можна умовно виділити:

– скінченно крокові методи;

– нескінченно крокові методи.

Скінченно кроковими (або кінцевими) називаються методи, що гарантують відшукання рішення задачі за кінцеве число кроків.

Для *нескінченно крокових* методів досягнення рішення гарантується лише в границях.

1.3. Збіжність методів оптимізації

Важливою характеристикою нескінченно крокових методів є *збіжність* [7].

Метод сходиться якщо $x^k \to x^*$ при $k \to \infty$, де x^* - розв'язок задачі.

Якщо $f(x^k) \to f(x^*)$, то іноді також кажуть, що метод сходиться (по функції), при цьому послідовність x^k називають мінімізуючою. Мінімізуюча послідовність може не збігатись до точки мінімуму.

У разі, коли точка мінімуму x^* не єдина, під збіжністю методу розуміється збіжність x^k до множини X^* точок мінімуму функції $f(x)$.

Нехай $x^k \to x^*$ при $k \to \infty$.

Кажуть, що:

1) послідовність x^k збігається до точки x^* лінійно (з лінійною швидкістю, зі швидкістю геометричної прогресії), якщо існують такі константи $q \in (0,1)$ і k_0, що

$$\left\|x^{k+1} - x^*\right\| \le \left\|x^k - x^*\right\| * q, \text{ при } k \ge k_0 \quad (1.3.1)$$

2) послідовність x^k збігається до точки x^* зверх лінійно (зі зверх лінійною швидкістю), якщо

$$\left\|x^{k+1} - x^*\right\| \le q_k * \left\|x^k - x^*\right\| \; q_k \to 0_+ \text{ при } k \to \infty \quad (1.3.2)$$

3) послідовність x^k збігається до точки x^* з квадратичною швидкістю збіжності, якщо істнують такі константи $\mathrm{c} \ge 0$ і k_0, що

$$\left\|x^{k+1} - x^*\right\| \le \mathrm{c} * \left\|x^k - x^*\right\|^2 \text{при } k \ge k_0 \quad (1.3.3).$$

Іноді, зберігаючи ту саму термінологію, нерівності (1.3.1) - (1.3.3) замінюють відповідно на нерівності

$$\left\|x^{k+1} - x^*\right\| \le \mathrm{c}_1 * q^{k+1} \text{ при } k \ge k_0. \quad (1.3.4)$$

$$\left\|x^{k+1} - x^*\right\| \le \mathrm{c}_2 * q_{k+1} * q_k * \ldots * q_1 \quad (1.3.5)$$

$$\left\|x^{k+1} - x^*\right\| \le \mathrm{c}_3 * (q^{k+1})^2 \text{ при } \; k \ge k_0, \, 0 \le q \le 1 \quad (1.3.6)$$

1.4. Умови зупинки розрахунку

Умова зупинки може визначатися наявними обчислювальними ресурсами.

На практиці часто використовують наступні умови зупинки:

$$\left\|x^{k+1} - x^k\right\| \le \varepsilon_1 \quad (1.4.1)$$

$$\left\|f(x^{k+1}) - f(x^k)\right\| \le \varepsilon_2 \quad (1.4.2)$$

$$\left\|f`(x^{k+1})\right\| \le \varepsilon_3 \quad (1.4.3)$$

Зазвичай користуються одною з умов, але іноді використовують критерії, що вимагають одночасного використання двох або усіх трьох умов (1.4.1) – (1.4.3).

1.5. Умови Оптимальності

Теорема 1. Якщо точка x^*досягає мінімуму диференціальної функції f(x):

$$f(x^*) = \min_x f(x), \tag{1.5.1}$$

то $f'(x^*)$=0. (1.5.2)

Визначимо похідну по напрямку:

$$f'(x,d) = \lim_{h\to+0}(f(x+hd) - f(x))/h, \tag{1.5.3}$$

яка характеризує швидкість росту функції f в точці x за напрямком d.

Елементарні обчислення показують, що для диференційованої функції $f(x)$ з похідною $f'(x)$ похідна за напрямком визначається формулою:

$$f'(x,d) = f'(x)d. \tag{1.5.4}$$

Припустимо $x = x^*$. Якщо $f'(x^*) \neq 0$, то

$$0 > -\|f'(x^*)\|^2 = -f'(x^*)f'(x^*) = f'\big(x^*, -f'(x^*)\big).$$

За визначенням (1.5.3) похідної за напрямком

$$f\big(x^* - hf'(x^*)\big) - f(x^*)/h = f'\big(x, -f'(x^*)\big) + O(h) = -\|f'(x^*)\|^2 + O(h),$$

де $O(h) \to 0$ при $h \to +0$.

Отже, для достатньо малих $h > 0$ залишковий член

$$|O(h)| < \|f'(x^*)\|^2/2$$

і відповідно

$$f\big(x^* - hf'(x^*)\big) - f(x^*) < -h\|f'(x^*)\|^2 + h\|f'(x^*)\|^2/2 = -\,h\|f'(x^*)\|^2/2 < 0$$

Якщо $f'(x^*) \neq 0$. Отже $f(x^*) - hf'(x^*)) < f(x^*)$ і x^* може бути точкою мінімуму.

Важливо, що, по-перше, вводить важливе поняття похідної за напрямком та, по-друге, визначає напрямок $(d = -f'(x)!)$, при зсуві в напрямі якого $(x \to x + hd = x - hf'(x)$ з точки x можна покращити (зменшити) цільову функцію, якщо її градієнт $f'(x)$ відмінний від нуля.

Перше дає можливість формулювати умови оптимальності і для інших класів функцій, тільки б у них існувала похідна за напрямом: прикладами таких функцій являються випуклі, але необов'язково диференційовані функції, частково-гладкі функції та інші.

Друге породжує градієнтний метод мінімізації, який досить ефективно використовується в спеціальних випадках.

З оптимальності точки x^* можна отримати інформацію і про поведінку її похідних в цій точці.

Теорема 2. Нехай точка x^* досягає мінімуму двічі диференційованої функції $f(x)$:

$$f(x^*) = \min_x f(x),$$

тоді матриця других похідних функції f в точці x^* визначена:

$$zf''(x^*)z \geq 0$$

для всіх z.

З оптимальності x^* слідує, що

$$f(x^*) \leq f(x^* + y) = f(x^*) + \frac{1}{2} yf''(x^*)y + o(\|y\|^2) \qquad (1.5.5)$$

Для всіх y, при чому

$o(\|y\|^2)/\|y\|^2 \to 0$ при $y \to 0$.

Розділивши нерівність (1.5.5) на $\|y\|^2$ та спрямувавши y до 0 так, щоб

y/ ||y||^2→ z,

отримаємо

$zf''(x^*)z \geq 0$.

Знову ж таки функція може мати в точці x^* невід'ємно визначену матрицю других похідних $f''(x^*)$, однак не мати в цій точці локального мінімуму.

В принципі, вона навіть може мати в цій точці максимум, як показує приклад $f(x) = -x^4$ $c\ x^* = 0$.

Легко бачити, що тверженная теореми 1 хибне: функція $f(x)$, зображена на рис.1 в точці x^* має нульову похідну, однак ця точка не являється ні мінімумом, ні максимумом $f(x)$.

Умови, гарантуючі оптимальність деякої точки, називається достатніми умовами екстремуму.

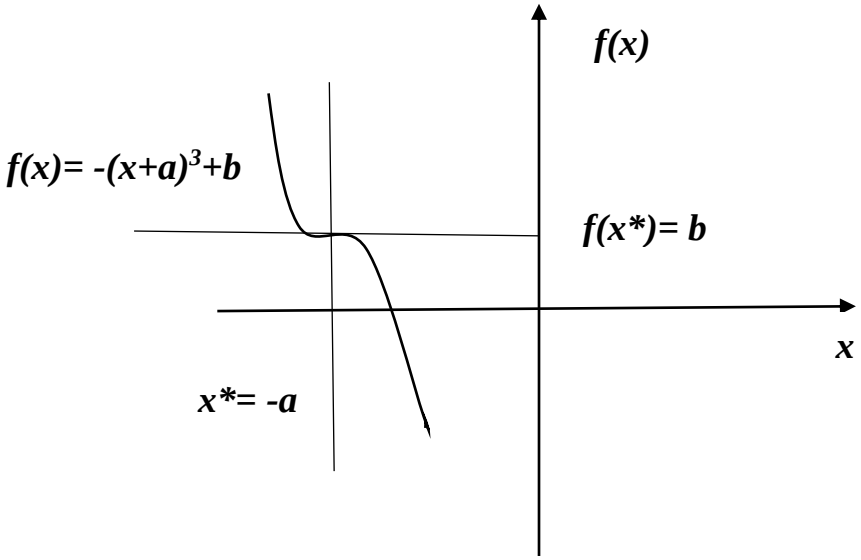


Рис. 1.5.1. Необхідні умови екстремуму не являються достатніми

Теорема 3. Якщо в точці x^* виконані умови:

1. $f'(x^*) = 0$,
2. Матриця других похідних $f''(x^*)$ позитивно визначена, то точка x^* являється ізольованою точкою локального мінімуму.
3. Ізольованість точки локального мінімуму x^* означає, що існує околиця точки x^* така що, у цій області при $x \neq x^*$ завжди $f(x) > f(x^*)$.

Позитивна визначеність $f''(x^*)$ означає, що

$zf''(x^*)z > 0$ для $z \neq 0$.

Позначимо

$$\gamma = \min_{\|z\|=1} zf''(x^*)z.$$

В силу замкненості та обмеженості єдиної сфери

$S = \{z: \|z\| = 1\}$,

а також в силу неперервності квадратичної функції $zf''(x^*)z$, як функція z, величина $\gamma > 0$.

Припустимо $x = x^* + \lambda z,\ z \in S$ та оцінимо

$$f(x) = f(x^* + \lambda z) = f(x^*) + \frac{1}{2}\lambda^2 zf''(x^*)z + o(\lambda^2),$$

Де $o(\lambda^2/\lambda^2 \to 0$ при $\lambda^2 \to 0$.

Звідси слідує, що

$$f(x) = f(x^*) + \frac{1}{2}\lambda^2\left(zf''(x^*)z + \frac{o(\lambda^2)}{\lambda^2}\right) \geq f(x^*) + \lambda^2\left(\frac{\gamma}{2} + \frac{o(\lambda^2)}{\lambda^2}\right)$$

$$\geq f(x^*) + \lambda^2\frac{\gamma}{4} = f(x^*) + \lambda^2\gamma$$

При цьому $\gamma' > 0$ для достатньо малих $\lambda > 0$. Тоді існує $\varepsilon > 0$ таке, що для всіх $0 < \|x - x^*\| = \lambda \leq \varepsilon$ виконується нерівність

$f(x) \geq f(x^*) + \gamma`\|x - x^*\|^2 > f(x^*)$

Та таким чином x^* являється ізольованою точкою локального мінімума.

Слабкість достатніх умов полягає в тому, що їх невиконання в точці x^* ще не каже про те, що ця точка не є точкою мінімуму.

Простим прикладом є функція $f(x) = x^4$. У точці $x^* = 0$ функція має ізольований навіть глобальний оптимум, однак достатні умови теореми 3 в точці x^*не виконані то що

$f``(x^*) = 12(x^*)^2 = 0.$

Питання до розділу 1.

1. Що називається оптимізацією?
2. Загальний вид задачі оптимізації.
3. Які є типи обмежень?
4. Типи задач оптимізації.
5. Групи методів оптимізації.
6. Різниця між локальним та глобальним екстремумом.

7. Що таке крок та ітерація методу.
8. Які є види збіжності методів?
9. Умови зупинки розрахунку
10. Необхідні умови оптимальності.
11. Достатні умови оптимальності.

РОЗДІЛ 2. ОПТИМІЗАЦІЯ ФУНКЦІЇ ОДНІЄЇ ЗМІННОЇ

2.1. Основні поняття

Функція $f(x)$ називається ***монотонною*** на проміжку $[a,b]$, якщо для двох довільних точок x_1 і x_2 із цього проміжку, таких що $x_1 < x_2$, виконується одна з нерівностей:

$f(x_1) \le f(x_2)$ - при зростанні функції,

$f(x_1) \ge f(x_2)$ - при спаданні функції.

Відзначимо, що монотонна функція не обов'язково повинна бути неперервною.

Якщо функція $f(x)$, визначена і неперервна на проміжку $[a,b]$, не є монотонною на цьому проміжку, то знайдуться такі частини $[a_1,b_1]$ проміжку $[a,b]$, в яких найбільше або найменше значення досягається функцією в точці, яка знаходиться між a_1 і b_1. На графіку функції (рис. 2.1.1) таким проміжкам відповідають характерні виступи або впадини.

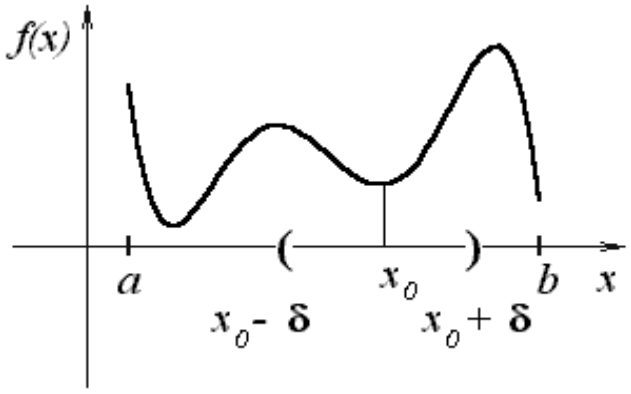


Рис. 2.1.1. Екстремуми функції однієї змінної

Кажуть, що функція $f(x)$ ***має*** в точці x_0 ***максимум (або мінімум)***, якщо цю точку можна оточити таким околом $(x_0-\delta, x_0+\delta)$, який повністю міститься в проміжку, на якому задана функція, що для будь-якої точки х цього околу виконується нерівність:

$f(x) \le f(x_0)$ – для точки максимуму,

$f(x) \ge f(x_0)$ – для точки мінімуму.

Для позначення максимуму і мінімуму використовують загальний термін ***екстремум***.

Якщо існує такий окіл, в межах якого (при $x \ne x_0$) виконується строга нерівність $f(x) < f(x_0)$ (або $f(x) > f(x_0)$), то кажуть, що функція має в точці x_0 власний максимум (мінімум).

Відомо [8; 14; 16], що диференційована функція $f(x)$ може мати екстремум в точках x_0, для яких $f'(x_0)=0$. Такі точки називають стаціонарними. Початковий етап розв'язування задачі на екстремум полягає в знаходженні всіх стаціонарних точок.

Функція $f(x)$ має екстремум в стаціонарній точці x_0, якщо в ній виконуються достатні умови існування екстремуму. При цьому, якщо в

стаціонарній точці $f''(x_0) > 0$, то маємо мінімум; якщо ж $f''(x_0) < 0$ - максимум.

Якщо друга похідна в стаціонарній точці не існує, то треба простежити за знаком першої похідної. Зміна знаку першої похідної при переході через стаціонарну точку з "-" на "+" вказує на наявність мінімуму. Зміна знаку першої похідної з "+" на "-" вказує на наявність максимуму. Такий спосіб знаходження екстремуму доцільно застосовувати в тих випадках, коли функція $f(x)$ і її похідні обчислюються порівняно просто.

Прості алгоритми для знаходження точок екстремуму можна побудувати для функції, яка є унімодальною (тобто, один екстремум) на відрізку $[a, b]$.

Функція $f(x)$ називається ***унімодальною*** на відрізку $[a, b]$, якщо на цьому відрізку існує точка x_0 така, що для будь-яких двох точок x_1 і x_2 цього відрізку, з умови $x_0 \le x_1 \le x_2$ слідує $f(x_0) \le f(x_1) \le f(x_2)$ або з умови $x_0 \ge x_1 \ge x_2$ слідує $f(x_0) \ge f(x_1) \ge f(x_2)$ (рис. 2.1.2).

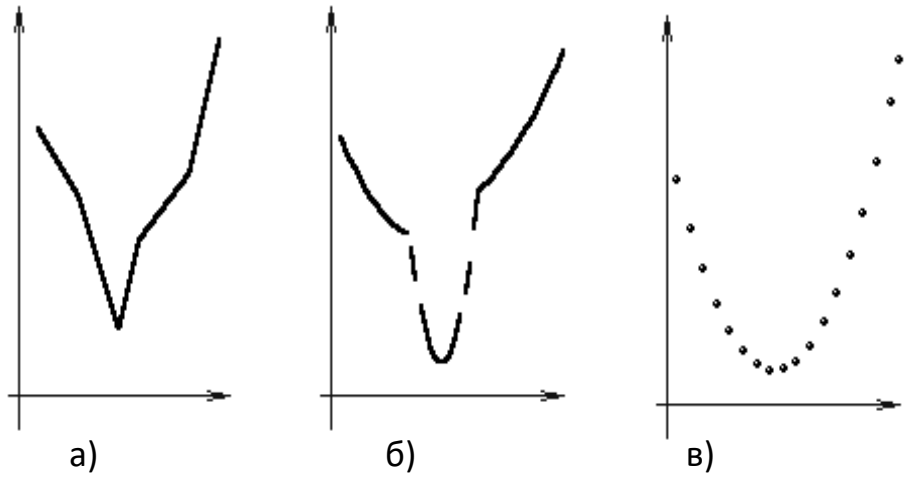


Рис. 2.1.2. Неперервна (а), розривна (б), дискретна (в) унімодальні функції

Іншими словами, унімодальна функція є монотонною по обидва боки від оптимальної точки. Унімодальні функції можуть бути неперервними, розривними, дискретними і іншими. Основною властивістю унімодальної

функції є те, що з допомогою будь-яких двох точок x_1, x_2 відрізку $[a,b]$ можна уточнити відрізок пошуку точки екстремуму [3; 5; 6].

2.2. Характеристика методів

В більшості випадків аналітичні методи визначення екстремуму функції неможливо використати у випадках, коли

- одержати аналітичний вираз для функції неможливо;
- аналітичний вираз для функції невідомий, але за допомогою деякого алгоритму може бути обчислене її значення для будь-якого значення аргументу;
- аналітичний вираз для функції невідомий і вона задана таблицею;
- значення похідних обчислюються досить складно.

В цих випадках застосовують числові методи. Такі методи розв'язування одновимірних задач оптимізації розділяють на наступні групи:

- методи виключення інтервалів;
- методи поліноміальної апроксимації;
- методи використання необхідних умов екстремуму.

До першої групи методів відносяться методи ділення навпіл (дихотомії), "золотого" ділення, Фібоначчі. Їх ідея – послідовне звуження інтервалу, який містить точку екстремуму [3; 5; 9]. Їх використовують, коли розв'язування рівняння $f'(x)=0$ є досить складним. В цих методах обчислюються лише значення функції в певних точках проміжку її визначення. При цьому доцільно прагнути, щоб при заданих вимогах до точності знаходження екстремуму кількість таких точок була якомога меншою. Послідовність дій при реалізації більшості із цих методів може бути такою:

- по деякому правилу вибирають кілька точок на відрізку $[a,b]$ і, виходячи з аналізу значень функції в цих точках, здійснюють локалізацію мінімуму (максимуму), тобто виділяють новий відрізок $[a_1,b_1]$ такий, що

$a \le a_1 \le b_1 \le b$, на якому продовжують процес пошуку екстремуму;

– на одержаному відрізку $[a_1, b_1]$ знову по вказаному правилу беруть ряд точок для наступної локалізації точки екстремуму, тобто для виділення відрізка $[a_2, b_2]$ такого, що $a_1 \le a_2 \le b_2 \le b_1$.

Процес продовжують, доки довжина чергового відрізку не стане меншою від заданої величини.

В методах першої групи для пошуку мінімуму функції $f(x)$, яка є унімодальною на відрізку $[a, b]$ використовують наступну теорему [2; 9].

Теорема. Нехай функція $f(x)$ є унімодальною на проміжку $a \le x \le b$ і її мінімум досягається в точці x_0. Тоді для будь-яких точок x_1 і x_2 відрізку $[a, b]$ таких, що $a < x_1 < x_2 < b$, справедливі наступні твердження:

якщо $f(x_1) > f(x_2)$, то точка мінімуму x_0 належить відрізку $[x_1, b]$. Відрізок $[a, x_1]$ можна виключити з розгляду;

якщо $f(x_1) < f(x_2)$, то точка мінімуму x_0 належить відрізку $[a, x_2]$. Відрізок $[x_2, b]$ можна виключити з розгляду (рис. 2.2.1).

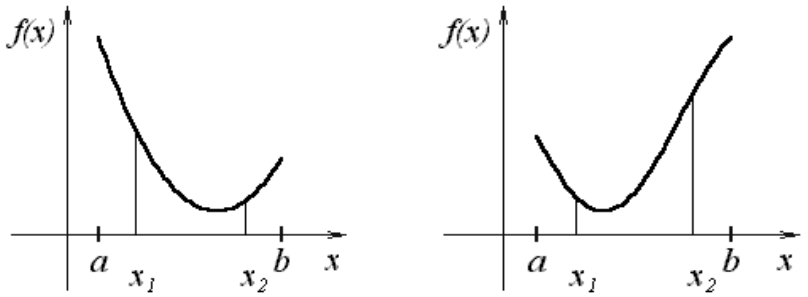


Рис. 2.2.1. Мінімум унімодальної функції

Зрозуміло, що якщо $f(x_1) = f(x_2)$, то можна відкинути обидва крайні інтервали і шукати точку мінімуму на відрізку $[x_1, x_2]$. Таким чином, ця теорема, яку ще називають правилом виключення інтервалів, дозволяє виключати з розгляду ті частини інтервалу $[a, b]$, на яких мінімум функції

відсутній.

Методи виключення інтервалів основані на простому порівнянні значень функції в двох пробних точках без урахування величини різниці між значеннями функції. Оптимізаційні алгоритми другої групи методів (поліноміальної апроксимації), в яких враховується відносна зміна значень функції, є більш ефективними. Але такий виграш в ефективності досягається за рахунок додаткової вимоги, згідно з якою досліджувана функція має бути достатньо гладкою. В курсі математичного аналізу [8] доводять теорему Вейєрштраса, зміст якої полягає в тому, що якщо функція $f(x)$ неперервна на деякому проміжку, то її з будь-якою точністю можна апроксимувати на цьому проміжку поліномом досить високого ступеню.

Основна ідея методів поліноміальної апроксимації полягає в заміні гладкої функції поліномом і використанні побудованого полінома для знаходження точки екстремуму.

Ясно, що якщо поліном досить точно апроксимує функцію $f(x)$ на деякому проміжку, то точка екстремуму полінома буде близькою до точки екстремуму функції. Збільшувати точність такого методу можна двома способами:

- використовувати поліном більш високого ступеню;
- застосовувати апроксимацію на меншому проміжку.

Через те, що побудова апроксимаційного полінома більш ніж третього ступеня є досить громіздкою процедурою, на практиці найчастіше використовують квадратичну - метод Пауелла, або кубічну апроксимацію [2; 8].

В методах третьої групи використовують необхідну умову існування екстремуму – рівність нулю першої похідної від функції $f(x)$. Тобто, задача зводиться до знаходження кореня рівняння $f'(x)=0$ одним з відомих методів розв’язування нелінійних алгебричних рівнянь, наприклад, хорд, Ньютона.

Розглянемо більш детально вказані методи.

2.3. Методи виключення інтервалів

2.3.1. Метод рівномірного пошуку

Розглянемо наступну задачу умовної оптимізації: знайти мінімум одновимірної унімодальної функції $f(x)$, визначеної в замкнутій області допустимих значень

$D = [a, b]$, $\min\limits_{x \in [a,b]} f(x) = f(x^*)$.

Ідея алгоритмів, що відносяться до методу скорочення поточного інтервалу невизначеності, полягає у виключенні у процесі пошуку з розгляду тих підінтервалів, у яких в силу унімодальності функції $f(x)$ точка x^* відсутня.

Позначимо поточний інтервал невизначеності як Δ, а його довжину | Δ |. Так що, якщо $\Delta = [a, b]$, то $|\Delta| = b - a$.

В алгоритмі рівномірного пошуку випробування проводяться в точках, які визначаються шляхом рівномірного розподілу інтервалу $[a, b]$ на N однакових підінтервалів. З обчислених значень функції $f(x)$ вибирається найменше. Нехай це значення досягається в точці x_k. Тоді в зв'язку з унімодальністю функції $f(x)$ підінтервали $[a, x_{k-1}], [x_{k+1}, b]$ можна виключити з розгляду, тобто зробити черговим інтервалом невизначеності інтервал $[x_{k-1}, x_{k+1}]$.

Алгоритм відноситься до класу пасивних методів пошуку.

Більш строго описану схему алгоритму можна записати в наступному вигляді.

1. Виконуємо присвоювання $r = 1, a^1 = a, b^1 = b, \Delta_1 = [a^1, b^1]$.
2. На поточному Δ будуємо рівномірну сітку з $N + 1$ вузлами.
3. Обчислюємо значення функції $f(x)$ у вузлах побудованої сітки $f(x_0^r), \dots, f(x_N^r)$.
4. Знаходимо мінімальне з цих значень:

$\min\big(f(x_0^r),\dots,f(x_N^r)\big)=f(x_k^r)$.

5. Виконуємо присвоювання $a^{r+1}=x_{k-1}^r, b^{r+1}=x_{k+1}^r, \Delta_{r+1}=[a^{r+1},b^{r+1}]$.

6. Якщо $|\Delta_{r+1}|\le\varepsilon_x$, тоді закінчуємо обчислення. Інакше – виконуємо присвоювання $r=r+1$ і переходимо на п.2. Тут ε_x - необхідна точність рішення.

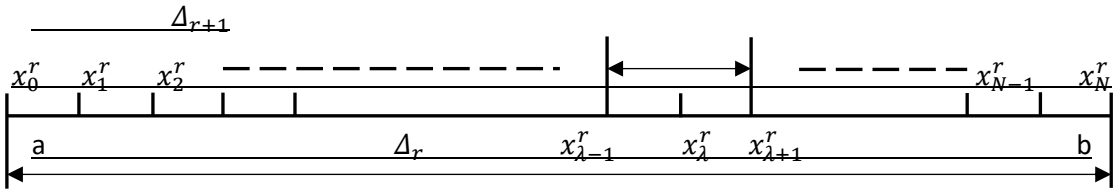


Рис 2.3.1.1. Побудова сітки на поточному інтервалі невизначеності.

Як наближене значення точки мінімуму x^* з рівними підставами може бути прийнята будь-яка точка останнього поточного інтервалу невизначеності.

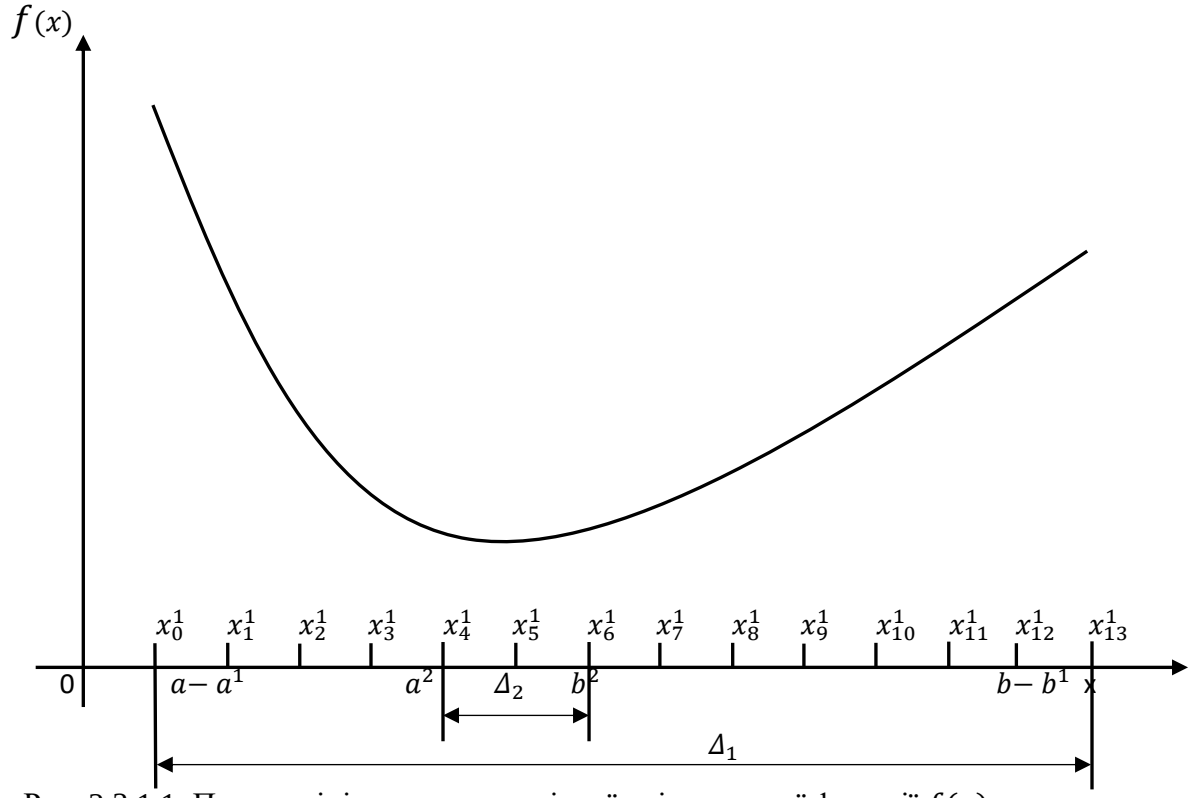


Рис. 2.3.1.1. Пошук мінімуму одновимірної унімодальної функції $f(x)$ за допомогою алгоритму рівномірного пошуку: $N=13$.

Першу ітерацію наведеної схеми алгоритму рівномірного пошуку ілюструє рис. 2.3.1.1.

Легко побачити, що після однієї ітерації алгоритму рівномірного пошуку Δ зменшується в $\frac{N}{2}$ раз. Тому кількість ітерацій r, необхідних для знаходження мінімуму функції з точністю ε_x , може бути знайдено з умови

$$\left(\frac{2}{N}\right)^r (b-a) \leq \varepsilon_x$$

2.3.2. Алгоритм ділення навпіл

Розглянемо наступну задачу умовної оптимізації: знайти мінімум одновимірної унімодальної функції $f(x)$, визначеної в замкнутій області допустимих значень

$f = [a, b]$, $\min\limits_{x \in [a,b]} f(x) = f(x^*)$.

В алгоритмі ділення навпіл або алгоритмі рівномірного дихотомічного пошуку випробування проводяться парами. Координати кожної наступної пари випробувань рознесені між собою на величину $\delta_x < \varepsilon_x$, де ε_x- необхідна точність рішення. Випробування виробляються в середині Δ. За значенням $f(x)$, отриманим в цих точках, одна половина Δ в силу унімодальності функції $f(x)$ виключається з подальшого розгляду.

Величина δ_x визначається необхідною точністю рішення. Алгоритм відноситься до класу методів послідовного пошуку.

Більш суворо описану схему алгоритму можна записати в наступному вигляді.

1. Виконуємо присвоювання $r = 1, a^1 = a, b^1 = b, \Delta_1 = [a^1, b^1]$.
2. Обчислити величини

$x_0^r = \frac{a^r - b^r}{2}, x_1^r = x_0^r - \frac{\delta_x}{2}, x_2^r = x_0^r + \frac{\delta_x}{2}$.

3. Обчислюємо значення $f(x_1^r), f(x_2^r)$ функції $f(x)$.

4. Якщо $f(x_1^r) < f(x_2^r)$, то виконуємо присвоювання $a^{r+1} = a^r, b^{r+1} = x_0^r, \Delta_{r+1} = [a^{r+1}, b^{r+1}]$. Інакше – виконуємо присвоювання $a^{r+1} = x_0^r, b^{r+1} = b^r, \Delta_{r+1} = [a^{r+1}, b^{r+1}]$.

5. Якщо $|\Delta_{r+1}| \leq \varepsilon_x$, тоді закінчуємо обчислення. Інакше – виконуємо присвоювання $r = r + 1$ і переходимо на п.2.

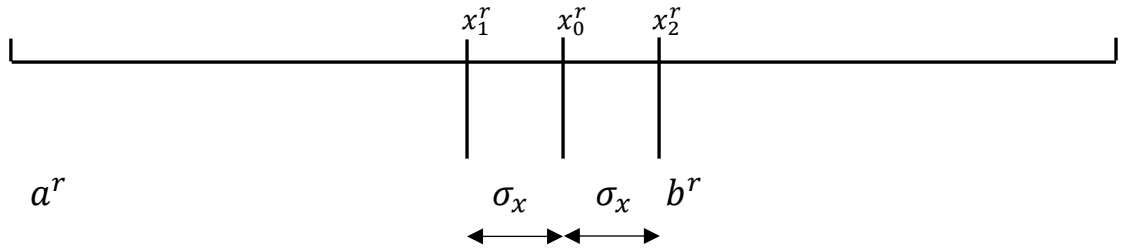


Рис. 2.3.2.1. До визначення величин x_0^r, x_1^r, x_2^r

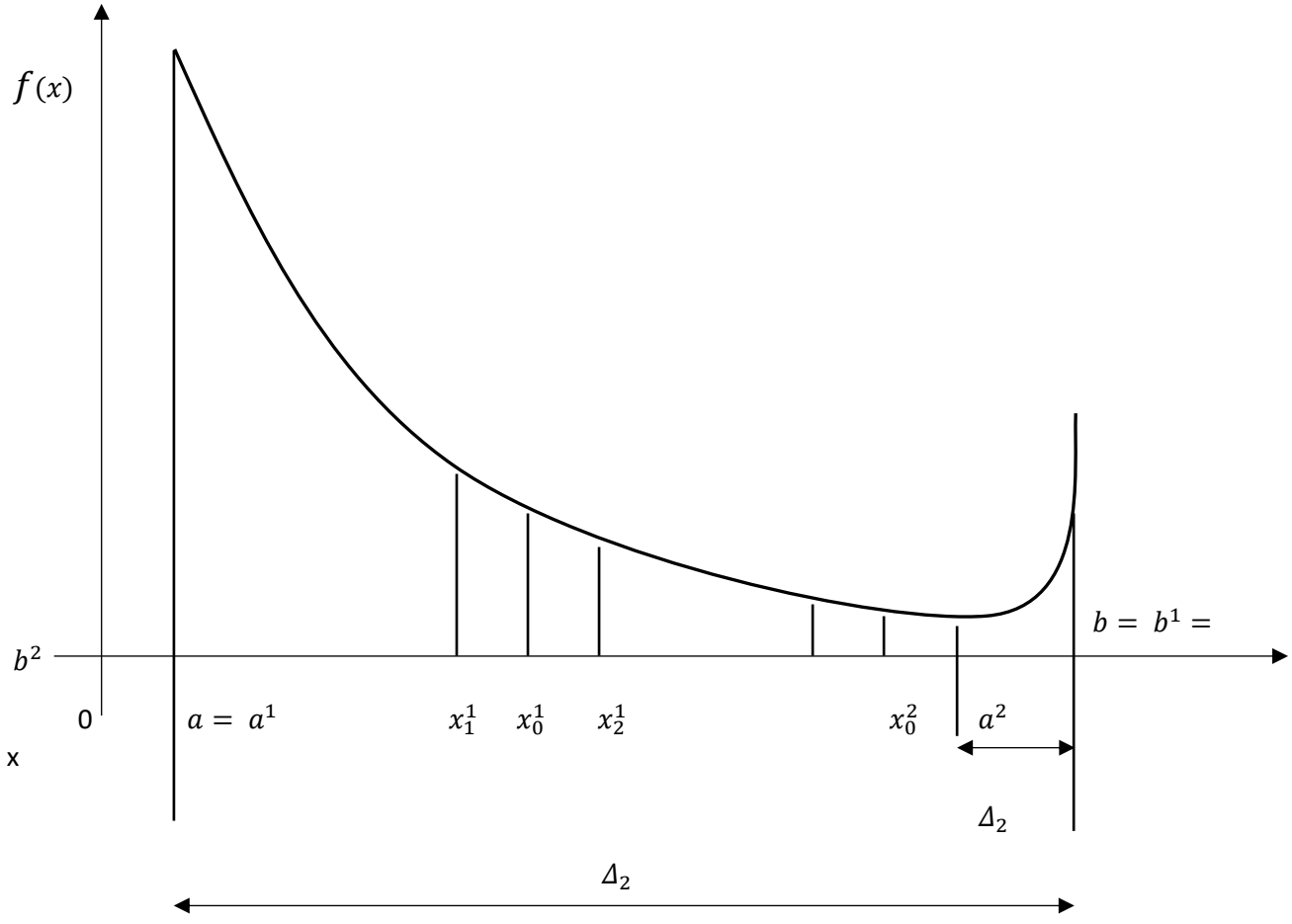


Рис. 2.3.2.2. Перші дві ітерації пошуку мінімуму одновимірної унімодальної функції за допомогою алгоритму рівномірного дихотомічного пошуку

Як наближене значення точки мінімуму x^* з рівними підставами може бути прийнята будь-яка точка останнього поточного інтервалу невизначеності.

Легко побачити, що після однієї ітерації алгоритму рівномірного пошуку Δ зменшується в 2 рази. Тому кількість ітерацій r, необхідних для знаходження мінімуму функції з точністю ε_x , може бути знайдено з умови

$\frac{b-a}{2^r} \leq \varepsilon_x$.

2.3.3. Метод Фібоначчі

Розглянемо наступну задачу умовної оптимізації: знайти мінімум одновимірної унімодальної функції $f(x)$, визначеної в замкнутій області допустимих значень

$D = [a, b]$,

$\min\limits_{x \in [a,b]} f(x) = f(x^*)$.

Числа Фібоначчі задаються наступним рекурентним рівнянням:

$i_N = i_{N-1} + i_{N-2}, N \geq 2, i_0 = i_1 = 1.$ (2.3.3.1)

Числа Фібоначчі $i_0, \dots, i_9$ наведені у наступній таблиці.

Таблиця 2.3.3.1.

N	0	1	2	3	4	5	6	7	8	9	…
i_N	1	1	2	3	5	8	13	21	34	55	…

Загальний вираз для N-го числа Фібоначчі можна отримати з розв’язку рівняння (1):

$$i_N = \frac{(\frac{1}{\tau})^{N+1} - (-\tau)^{N+1}}{\sqrt{5}}, \text{где } \tau = \frac{\sqrt{5}-1}{2} \approx 0.618$$

При великих значеннях N членом $(-\tau)^{N+1}$ можна знехтувати. При цьому

$\mathrm{i_N} \approx \frac{(\frac{1}{\tau})^{\mathrm{N+1}}}{\sqrt{5}}$. (2.3.3.2)

Звідси випливає, що $\frac{i_{N-1}}{i_N} \approx \tau$. Тобто відношення двох сусідніх чисел Фібоначчі приблизно постійне і дорівнює τ.

Алгоритм методу Фібоначчі.

Алгоритм Фібоначчі належить до класу пошукових методів оптимізації і включає в собі 2 етапи.

Перший етап складається $(N-1)$-ї ітерації для $r = 1,2,\dots,N-1$.

Розглянемо схему r-ї ітерації, коли $\Delta_r = [a^r, b^r]$:

1. Обчислюємо

$x_1^r = a^r + |\Delta_r| \frac{i_{N-1-r}}{i_{N+1-r}}$,

$x_2^r = a^r + |\Delta_r| \frac{i_{N-r}}{i_{N+1-r}}$.

2. Обчислюємо значення $f(x_1^r), f(x_2^r)$ функції $f(x)$.

3. Якщо $f(x_1^r) < f(x_2^r)$, тоді виконуємо присвоювання $a^{r+1} = a^r, b^{r+1} = x_2^r, \Delta_{r+1} = [a^{r+1}, b^{r+1}]$. Інакше – виконуємо присвоювання $a^{r+1} = x_1^r, b^{r+1} = b^r, \Delta_{r+1} = [a^{r+1}, b^{r+1}]$.

Алгоритм Фібоначчі володіє такою властивістю, що після виконання $(N-1)$-ї ітерації має місце наступна ситуація: $x_1^{N-1} = x_2^{N-1} = x^{N-1}$. Тобто в результаті $(N-1)$-ї ітерації звуження поточного інтервалу невизначеності не відбувається:

$\Delta_{N-1} = [a^{N-1}, b^{N-1}] = \Delta_{N-2} = [a^{N-2}, b^{N-2}]$.

Другий етап покликаний вирішити по який бік від точки x^{N-1} лежить точка мінімуму функції $f(x)$.

Другий етап виконується за наступною схемою:

1. Знаходимо точку $x^N = x^{N-1} + \sigma_x$, де $\sigma_x \ll |\Delta_{N-1}|$- вільний параметр алгоритму.

2. Обчислюємо значення функції $f(x^N)$.

3. Якщо $f(x^N) > f(x^{N-1})$, тоді виконуємо присвоювання $\Delta_N = [a^{N-1}, x^{N-1}]$. Інакше – виконуємо присвоювання $\Delta_N = [x^{N-1}, b_N]$.

Як наближене значення точки мінімуму x^* з рівними підставами може бути прийнята будь-яка точка Δ_N.

Тому кількість ітерацій N, необхідна для знаходження мінімуму функції з точністю ε_x, знаходиться з умови

$$\varepsilon_x \leq \frac{b-a}{i_N}$$

2.3.4. Алгоритм золотого перетину

Розглянемо наступну задачу умовної оптимізації. Треба знайти мінімум одновимірної унімодальної функції $f(x)$, визначеної в замкнутої області допустимих значень

$D = [a, b]$, $\min\limits_{x \in [a,b]} f(x) = f(x^*)$.

Властивості золотого перетину

Розглянемо інтервал $[a, b]$. Точка c виконує золотий перетин інтервалу $[a, b]$, якщо

$$\frac{c-a}{b-a} = \tau\ , \qquad (2.3.4.1)$$

де $\tau = \frac{\sqrt{5}-1}{2} \approx 0{,}618$ – розв'язок квадратного рівняння $\tau^2 + \tau - 1 = 0$. (2.3.4.2)

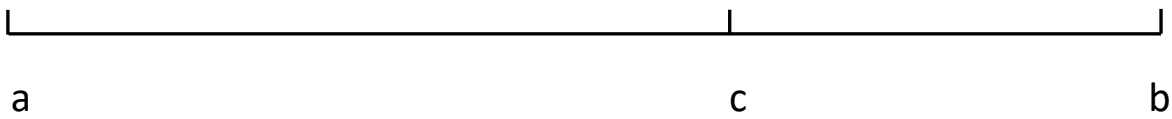


Рис. 2.3.4.1. До визначення золотого перетину відрізка

З означення золотого перетину слідує, що $\frac{c-a}{b-a} = 1 - \tau$.

Дійсно, $\frac{b-c}{b-a} = \frac{(b-a)-(c-a)}{b-a} = 1 - \frac{c-a}{b-a} = 1 - \tau$.

Алгоритм золотого перетину належить до класу послідовних методів пошуку.

Виконаємо присвоювання $r = 1, a^1 = a, b^1 = b, \Delta_1 = [a^1, b^1]$.

1. Обчислюємо величини

$$x_1^r = b^r - (b^r - a^r)\tau, x_2^r = a^r + (b^r - a^r)\tau. \quad (2.3.4.3)$$

2. Обчислюємо значення $f(x_1^r), f(x_2^r)$ функції $f(x)$.

3. Якщо $f(x_1^r) < f(x_2^r)$, тоді виконуємо присвоювання

$a^{r+1} = a^r, b^{r+1} = x_2^r,$

$\Delta_{r+1} = [a^{r+1}, b^{r+1}].$

4. Інакше – виконуємо присвоювання

$a^{r+1} = x_1^r, b^{r+1} = b^r,$

$\Delta_{r+1} = [a^{r+1}, b^{r+1}].$

5. Якщо $|\Delta_{r+1}| \leq \varepsilon_x$, тоді закінчуємо обчислення. Інакше – виконуємо присвоювання $r = r + 1$і переходимо на пункт 2.

Тут ε_x - необхідна точність рішення.

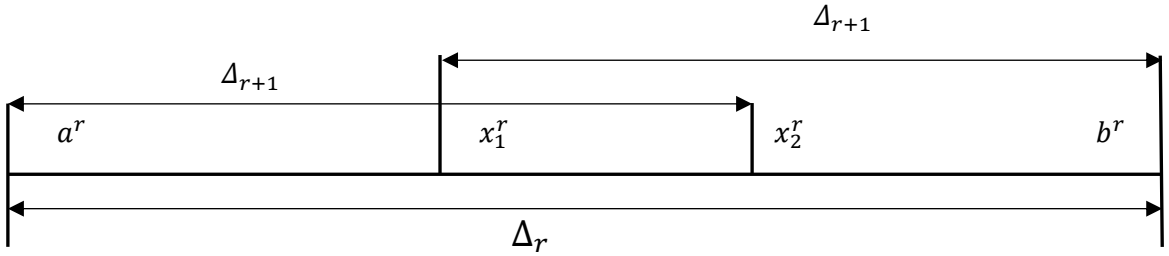


Рис. 2.3.4.2. Визначення величин x_1^r, x_2^r

Як наближене значення точки мінімуму x^* з рівними підставами може бути прийнята будь-яка точка останнього поточного інтервалу невизначеності.

Особливості алгоритму золотого перетину:

1. Точки x_1^r, x_2^r розташовані симетрично щодо кінців поточного інтервалу невизначеності.

Дійсно, з (2.3.4.3) слідує, що точка x_1^r відстає від точки b^r на величину $(b^r - a^r)\tau$; точка x_2^{r+1} відстає від точки a^r на таку саму величину.

2. Для будь-якого $r \geq 1$ алгоритм золотого перетину володіє наступною властивістю: одна з точок x_1^{r+1}, x_2^{r+1} збігається з однією з точок x_1^r, x_2^r.

Вказана властивість алгоритму золотого перетину дозволяє на кожній ітерації (окрім першої) проводити випробування тільки в одній точці.

3. У результаті однієї ітерації алгоритму золотого перетину довжина поточного інтервалу невизначеності скорочується в τ разів.

Тому кількість ітерацій N, необхідна для знаходження мінімуму функції з точністю ε_x, знаходиться з умови $\varepsilon_x \leq (b-a)\tau^N$.

2.3.5. Порівняння ефективності алгоритмів одновимірної оптимізації

Визначення критерію якості алгоритмів оптимізації можливо тільки для наступної одновимірної задачі умовної оптимізації. Знайти мінімум одновимірної унімодальної функції $f(x)$, визначеної в замкнутій області допустимих значень

$$D = [a,b],\ \min_{x\,\epsilon\,[a,b]} f(x) = f(x^*). \qquad (2.3.5.1)$$

Нехай розглядаються наступні алгоритми:

A_1- алгоритм рівномірного пошуку,

A_2- алгоритм рівномірного дихотомічного пошуку,

A_3- алгоритм Фібоначчі,

A_4- алгоритм золотого перетину.

В якості критерію оптимальності алгоритмів використовуємо максимальну довжину поточного інтервалу невизначеності після N випробувань.

Кількість вузлів сітки рівна $N+1$, тоді після однієї ітерації алгоритму рівномірного пошуку поточний інтервал невизначеності зменшується у $\frac{N}{2}$ разів. Тому

$K(A_1) = \frac{2(b-a)}{N-1}$.

Після однієї ітерації алгоритму рівномірного пошуку поточний інтервал невизначеності зменшується в 2 рази. Тому

$K(A_2) = \frac{(b-a)}{2^{N/2}}$.

В результаті N ітерацій ($N+1$ випробувань) алгоритму Фібоначчі довжина поточного інтервалу невизначеності стає рівна $\frac{b-a}{i_N}$. Тоді

$K(A_3) = \frac{b-a}{i_{N-1}}$.

В результаті N ітерацій ($N+1$ випробувань) алгоритму золотого перетину довжина поточного інтервалу невизначеності стає рівна $(b-a)\tau^N$. Тому

$K(A_4) = (b-a)\tau^{N-1}$.

Порівняємо ефективності алгоритму ділення навпіл і алгоритму Фібоначчі при $N = 14$:

$$\frac{K(A_2)}{K(A_3)} = \frac{\frac{(b-a)}{2^{N/2}}}{\frac{b-a}{i_{N-1}}} = \frac{i_{N-1}}{2^{N/2}} = \frac{377}{128} \approx 3.$$

Таким чином, при $N = 14$ алгоритм Фібоначчі майже в 3 рази ефективніший за алгоритм ділення навпіл.

При $N = 14$ порівняємо також ефективність алгоритму Фібоначчі та алгоритму золотого перетину:

$$\frac{K(A_3)}{K(A_4)} = \frac{1}{i_{N-1}\tau^{N-1}} \approx \frac{\tau^N\sqrt{5}}{\tau^{N-1}} = \tau^N\sqrt{5} \approx 1.4.$$

Тут враховано, що $i_N \approx \frac{(1/\tau)^{N+1}}{\sqrt{5}}$.

Таким чином, при $N = 14$ алгоритм золотого перетину приблизно на 40 відсотків ефективніший за алгоритм Фібоначчі.

Отже, загальними рисами прямих методів є:

– можливість використовувати їх для аналізу як неперервних, так і розривних і дискретних функцій;

– логічна структура пошуку основана на простому порівнянні значень функції в двох пробних точках;

– ефективність методів можна порівняти наступним чином. Якщо N – кількість ітерацій, то відносне зменшення інтервалу пошуку для розглянутих методів визначається за формулами:

$\left(\frac{1}{2}\right)^N$ – для методу дихотомії,

$(\tau)^N$ – для методу “золотого” ділення,

$\frac{1}{F_N}$ – для методу Фібоначчі;

– використання методів виключення інтервалів накладає на досліджувану цільову функцію єдине обмеження: функція має бути унімодальною. Але це обмеження не є принциповим.

2.4. Методи поліноміальної апроксимації

2.4.1. Помилка! Закладку не визначено.Метод Пауелла (квадратичної апроксимації)

Нехай задані точки $x_1 < x_2 < x_3$ і відомі значення функції $f(x)$ в цих точках. Тоді можна визначити числа A_0, A_1, A_2 так, що значення полінома другого ступеня виду

$$P(x) = A_0 + A_1(x - x_1) + A_2(x - x_1)(x - x_2)$$

в точках x_1, x_2, x_3 співпадатимуть з значеннями функції $f(x)$ в цих точках [6]. Дійсно,

$$f_1 = f(x_1) = P(x_1) = A_0;$$

$$f_2 = f(x_2) = P(x_2) = f_1 + A_1(x_2 - x_1);\ A_1 = \frac{f_2 - f_1}{x_2 - x_1};$$

$$f_3 = f(x_3) = P(x_3) = f_1 + \frac{f_2 - f_1}{x_2 - x_1}(x_3 - x_1) + A_2(x_3 - x_1)(x_3 - x_2);$$

$$A_2 = \frac{1}{x_3 - x_2}\left[\frac{f_3 - f_1}{x_3 - x_1} - \frac{f_2 - f_1}{x_2 - x_1}\right].$$

Отже, можна побудувати поліном другого ступеня, який завжди є унімодальною функцією і апроксимує функцію $f(x)$ на досліджуваному інтервалі. Якщо функція $f(x)$ є унімодальною на цьому інтервалі, то побудований поліном можна використати для оцінки значення точки екстремуму функції $f(x)$, виконавши стандартні дії:

$$\frac{dP}{dx} = A_1 + A_2(x - x_2) + A_2(x - x_1) = 0;\ \bar{x} = \frac{x_2 + x_1}{2} - \frac{A_1}{2A_2}.$$

Алгоритм методу Пауелла пошуку мінімуму має вигляд:

Крок 0. Вибрати вихідну точку x_1 і крок *h*.

Крок 1. Обчислити $x_2 = x_1 + h,\ f(x_1),\ f(x_2)$.

Крок 2. Якщо $f(x_1) > f(x_2)$ то $x_3 = x_1 + 2h$; якщо $f(x_1) < f(x_2)$ то $x_3 = x_1 - h$.

Крок 3. Обчислити $f(x_3)$. Знайти $F_{\min} = \min(f_1, f_2, f_3)$. Величині $x_{\min}$ присвоїти значення відповідного аргументу.

Крок 4. По точках $x_1,\ x_2,\ x_3$ побудувати квадратичну апроксимацію і знайти $\bar{x}$, використовуючи формули мінімуму квадратного тричлена.

Крок 5. Якщо $\left|F_{\min} - f(\bar{x})\right| < \varepsilon$ або $\left|x_{\min} - \bar{x}\right| < \varepsilon$, то вихід з алгоритму.

Крок 6. Вибрати "найкращу" точку із $\{x_{\min}, \bar{x}\}$ і дві точки по обидва боки від неї. Перенумерувати точки $x_1,\ x_2,\ x_3$ і перейти на Крок 3.

2.4.2. Метод кубічної апроксимації

Нехай відомо, що функція $f(x)$ є унімодальною на деякому проміжку. Знайдемо на цьому проміжку точки x_1 і x_2 такі, що похідні $f'(x_1)$ і $f'(x_2)$ мають в цих точках різні знаки. Тоді точка екстремуму функції $f(x)$ знаходиться між x_1 і x_2. Апроксимуємо функцію $f(x)$ поліномом третього ступеня виду

$$P(x) = A_0 + A_1(x - x_1) + A_2(x - x_1)(x - x_2) + A_3(x - x_1)^2(x - x_2).$$

Коефіцієнти A_0, A_1, A_2, A_3 знаходимо з умови, що значення полінома $P(x)$ і його перших похідних в точках x_1 і x_2 співпадають з значеннями функції $f(x)$ і її похідними в цих точках [6]. Тобто, повинні виконуватися умови

$$\begin{cases} P(x_1) = f(x_1), \\ P(x_2) = f(x_2), \\ P'(x_1) = f'(x_1), \\ P'(x_2) = f'(x_2). \end{cases}$$

Оскільки

$$\frac{dP}{dx} = A_1 + A_2(x - x_1) + A_2(x - x_2) + A_3(x - x_1)^2 + 2A_3(x - x_1)(x - x_2),$$

то для знаходження коефіцієнтів A_i одержимо систему чотирьох рівнянь, яка легко розв'язується рекурсивним методом.

$$\begin{cases} f(x_1) = A_0, \\ f(x_2) = A_0 + A_1(x_2 - x_1), \\ f'(x_1) = A_1 + A_2(x_1 - x_2), \\ f'(x_2) = A_1 + A_2(x_2 - x_1) + A_3(x_2 - x_1)^2. \end{cases}$$

Для знаходження стаціонарних точок апроксимуючого полінома одержимо квадратне рівняння. Для того, щоб із двох коренів квадратного рівняння відразу знайти потрібний корінь, треба виконати наступні дії. Послідовно знаходимо числа:

$$Z = \frac{3(f_1 - f_2)}{x_2 - x_1} + f'_1 + f'_2; \qquad W = \begin{cases} \sqrt{Z^2 - f'_1 f'_2} & , при \quad x_1 < x_2, \\ -\sqrt{Z^2 - f'_1 f'_2} & , при \quad x_1 > x_2. \end{cases}$$

$$m = \frac{f'_2 + W - Z}{f'_2 - f'_1 + 2W}.$$

Потрібну точку $\bar{x}$ знаходимо по правилу:

$$\bar{x} = \begin{cases} x_2 & m < 0; \\ x_2 - m(x_2 - x_1) & 0 \le m \le 1; \\ x_1 & m > 1. \end{cases}$$

Тепер треба оцінити значення функції $f(x)$ і її похідної $f'(x)$ в

одержаній точці $\bar{x}$. Якщо в точці $\bar{x}$ похідна досить мала, то ітераційний процес закінчується. Якщо ж величина похідної ще залишається великою, то треба знайти точку x_T , таку, що $f'(x_T)f'(\bar{x}) < 0$, і повторити обчислення на інтервалі $[x_T, \bar{x}]$.

2.5. Методи використання умов екстремуму

Розглянемо наступну задачу оптимізації: знайти мінімум одномірної унімодальної функції *f(x)*, визначеної у замкнутій області допустимих значень *D = [a,b],*

$$\min f(x) = f(x^*) \qquad (2.5.1)$$

$$x \in [a, b]$$

Раніше ми показали, що в цих припущеннях необхідною умовою мінімуму функції *f(x)* є умова

$$f'(x)=0,\ x \in [a,b]. \qquad (2.5.2)$$

Розглядаються методи, що базуються на пошуку стаціонарної точки функції *f(x),* тобто на розв’язанні задачі (2.5.2), яка представляє собою задачу знаходження коренів функції *f(x)* , що належать інтервалу *[a b].*

Аналітичний розв’язок задачі (2.5.2) можливий лише у найпростіших випадках. Зазвичай для розв’язання цієї задачі доводиться використовувати чисельні методи знаходження коренів нелінійних рівнянь.

Широко відомі наступні методи знаходження коренів нелінійних рівнянь:

– метод хорд (метод січних);

– метод дотичних (метод Ньютона розв’язку нелінійних рівнянь).

2.5.1. Метод хорд

Метод хорд [10; 15] орієнтований на знаходження коренів рівняння (2.5.2) у випадку, коли на межах інтервалу *[a,b]* знаки похідної *f′(x)* відрізняються. Така ситуація, очевидно, можлива, якщо точка мінімуму функції *f(x)* є внутрішньою точкою інтервалу *[a,b]* – див. рис. 2.5.1.1.

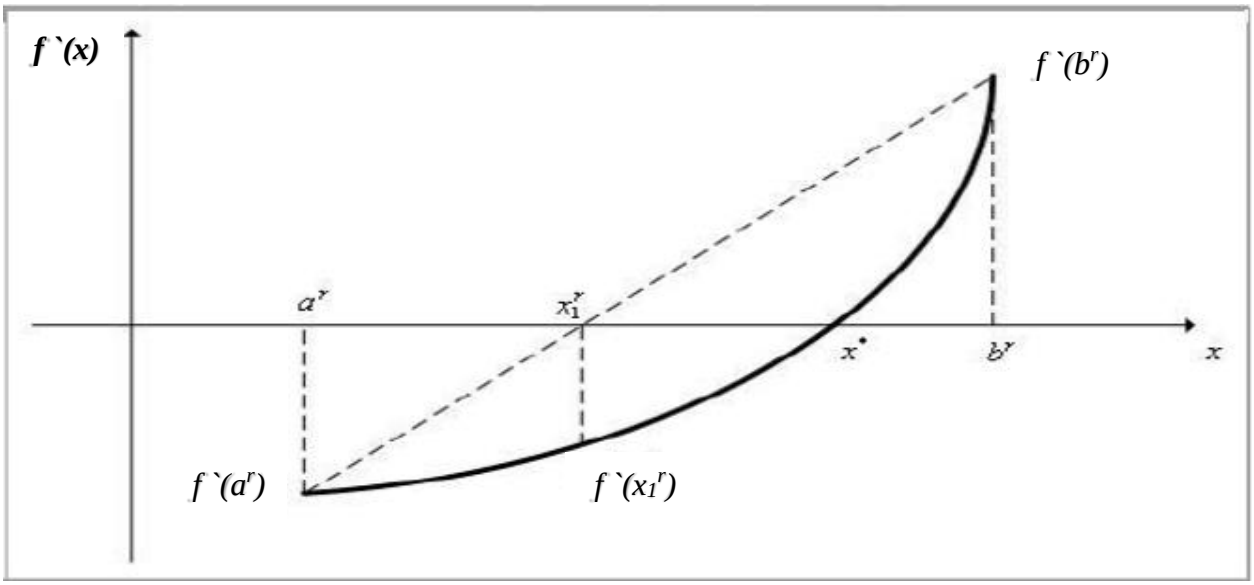


Рис. 2.5.1.1. До схеми метода хорд

Далі нам знадобиться значення x_1^r. З подоби трикутника a^r, x_1^r, $f'(a^r)$ і трикутника b^r, x_1^r, $f''(b^r)$ маємо

$$\frac{f'(b^r)}{f'(a^r)} = \frac{b^r - x_1^r}{x_1^r - a^r}$$

Звідси слідує, що

$$x_1^r = b^r - \frac{f'(b^r)(b^r - a^r)}{f'(b^r) - f'(a^r)} \qquad (2.5.1.1)$$

Схема *пошуку стаціонарної точки мінімізуємої функції методом хорд*:

1. Виконуємо присвоювання $r=1$, $a^1=a$, $b^1=b$
2. Обчислюємо значення похідних $f'(a^r)$, $f'(b^r)$.
3. Якщо похідні $f'(a^r)$, $f''(b^r)$ мають однакові знаки – завершуємо обчислення (точки a, b обрані невірно).

За формулою (2.5.1.1) обчислюємо наближення x_1^r до стаціонарної точки функції $f(x)$ і значення похідної $f'(x_1^r)$.

4. Якщо $|f'(x_1^r)| \le \varepsilon$, де ε - необхідна точність розв'язку, то приймаємо $x^* \approx x_1^r$ і завершуємо обчислення.
5. Якщо похідні $f'(a^r)$, $f'(x_1^r)$ мають різні знаки, то виконуємо присвоювання $a^{r+1}=a^r$, $b^{r+1}= x_1^r$, $r=r+1$ і переходимо на пункт 4.

6. Якщо похідні $f'(x_1^r)$, $f'(b^r)$ мають різні знаки (як на рис. 2.5.1.1), то виконуємо присвоювання. $a^{r+1}= x_1^r$, $b^{r+1}=b^r$, $r=r+1$ і переходимо на пункт 4.

У випадку квадратичної функції $f(x)$ похідна цієї функції $f'(x)$ лінійна. Тому метод хорд гарантує знаходження стаціонарної точки функції $f(x)$ всього за одну ітерацію.

Оскільки пошук закінчується при виконанні умови $|f'(x_1^r)| \leq \varepsilon$, можлива поява помилкових коренів. Наприклад, для рівняння $x^2+0.0001=0$ помилковий корінь $x=0$ з'являється у тому випадку, якщо $\varepsilon > 0.0001$. У подібних випадках збільшуючи точність пошуку, можна позбавитися помилкових коренів. Втім, можливі рівняння, для яких такий підхід не приводить до успіху. Наприклад, рівняння $\frac{1}{x} = 0$ не має дійсних коренів, втім для скільки завгодно малого ε знайдеться точка, що задовольнятиме умові закінчення пошуку.

Можлива модифікація метода хорд, коли значення похідної $f'(x)$ обчислюються наближено з використанням перших різниць. В цьому випадку, очевидно, метод стає прямим (нульового порядку).

2.5.2. Пошук стаціонарної точки методом дотичних (Метод Ньютона)

Метод дотичних [10; 15] орієнтований на знаходження кореня рівняння (2.5.2) у випадку, коли на межах інтервалу $[a, b]$ знаки похідної $f'(x)$ відрізняються. Така ситуація, очевидно, можлива, якщо точка мінімуму функції $f(x)$ є внутрішньою точкою інтервалу $[a, b]$ – див. рис. 2.5.2.1., метод потребує, щоб функція $f(x)$ була визначена і двічі диференційована в області допустимих значень $D=[a, b]$.

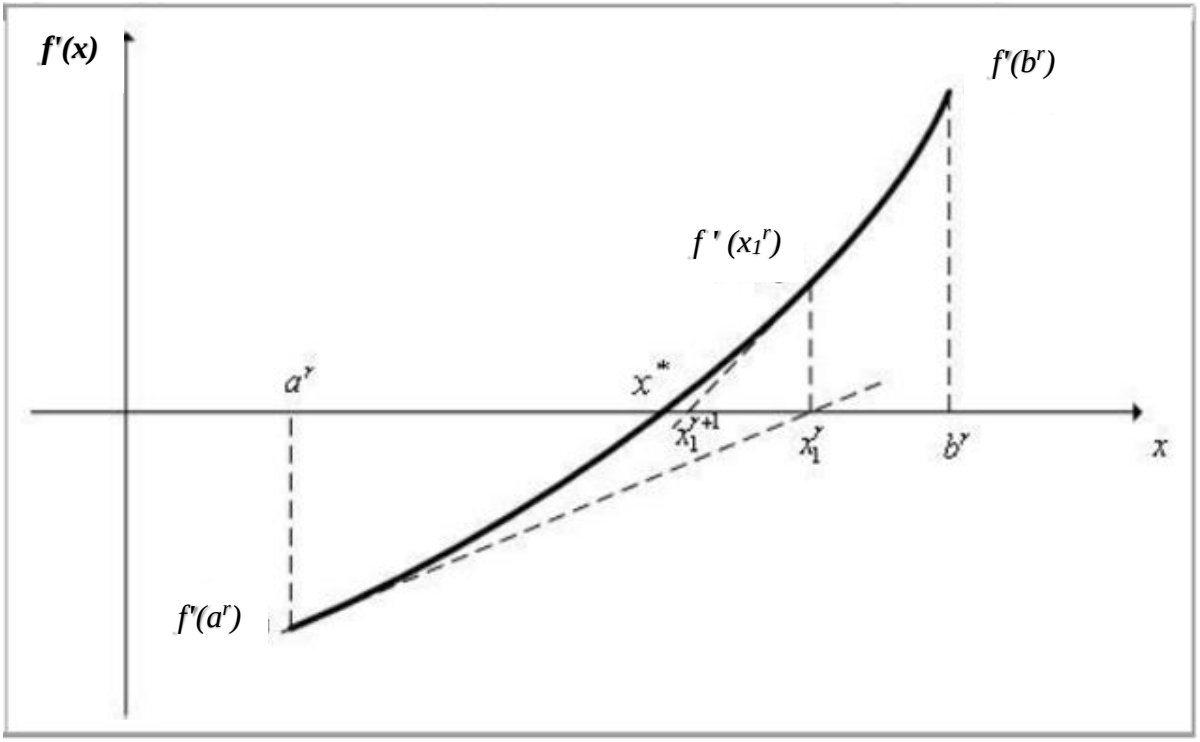


Рис. 2.5.2.1. До схеми методу дотичних

Далі нам знадобиться значення x_1^r. Лінійна функція, апроксимуюча $f'(x)$ у точці a^r записується у вигляді

~,

$$f(x, a^r) = f'(a^r) + f''(a^r)(x - a^r) \qquad (2.5.2.1)$$

Прирівнявши праву частину рівняння (2.5.2.1) до нуля, отримаємо

$$x_1^r = a^r - \frac{f'(a^r)}{f''(a^r)} \qquad (2.5.2.2)$$

Схема пошуку стаціонарної точки мінмізуємої функції методом дотичних:

1. Виконуємо присвоювання $r=1$, $a^1=a$, $b^1=b$
2. Обчислюємо значення похідних $f'(a^r)$, $f'(b^r)$.
3. Якщо похідні $f'(a^r)$, $f'(b^r)$ мають однакові знаки – завершуємо обчислення (точки a, b обрані невірно).
4. За формулою (2.5.2.2) обчислюємо наближення x_1^r до стаціонарної точки функції $f(x)$ і значення похідної $f'(x_1^r)$.

5. Якщо $|f'(x_1^r)| \le \varepsilon$, де ε - необхідна точність розв'язку, то приймаємо $x^* \approx x_1^r$ і завершуємо обчислення.

6. Якщо різні знаки мають похідні $f'(a^r)$, $f'(x_1^r)$, то виконуємо присвоювання $a^{r+1}=a^r$, $b^{r+1}= x_1^r$, $r=r+1$ і переходимо на п.4.

7. Якщо різні знаки мають похідні $f'(x_1^r)$, $f'(b^r)$, то виконуємо присвоювання. $a^{r+1}= x_1^r$, $b^{r+1}=b^r$, $r=r+1$ і переходимо на п.4.

У випадку квадратичної функції $f(x)$ похідна цієї функції $f'(x)$ лінійна. Тому метод дотичних гарантує знаходження стаціонарної точки функції $f(x)$ всього за одну ітерацію.

Також, як і в методі хорд, можлива модифікація метода дотичних, коли значення похідної $f'(x)$ обчислюються наближено з використанням перших різниць. В цьому випадку, очевидно, метод стає прямим (нульового порядку).

2.5.3. Підвищення ефективності пошуку на основі умови Ліпшица

Розглянемо задачу умовної оптимізації: треба знайти мінімум одномірної унімодальної функції $f(x)$, визначеної у замкнутій області допустимих значень $D = [\, a\, ,\, b\,]$,

$$\min f(x) = f(x^*) \qquad (2.5.3.1)$$

$$x \in [a, b]$$

Усяка апріорна інформація про властивості мінімізуємої функції $f(x)$ може бути використана для підвищення ефективності розв'язання задачі пошуку мінімуму функції (2.5.3.1).

Припустимо, що є наступна апріорна інформація про мінімізуєму функцію: $f(x)$ є ліпшицевою функцією, тобто належить до класу функцій, які на інтервалі $[a, b]$ задовольняють умову Ліпшица [11; 12] з константою Ліпшица К

$$|f(x_1)- f(x_2)| \le K\,|x_1-x_2|,\ \forall\ x_1, x_2 \in [a, b]$$

$$K = \frac{|f(x1)- f(x2)|}{|x1-x2|}.$$

Розглянемо, як можна використовувати апріорну інформацію для скорочення інтервалу невизначеності без проведення додаткових досліджень функції.

Після проведення r випробувань яким-небудь методом скорочення поточного інтервалу невизначеності має місце ситуація:

$a^r=x_1^r < x_2^r < b^r = x_3^r,\ f(a^r) > f(x_1^r) > f(b^r)$, $[a^r, b^r]$.

Позначимо $f\left(x_j^r\right) = f_j^r$, j=1,2,3 і проведемо через точки (x_j^r, f_j^r), j=1,3 прямі L_1, L_3 з тангенсами кута нахилу до осі X, рівними K і (-K), відповідно (рис. 2.5.3.1).

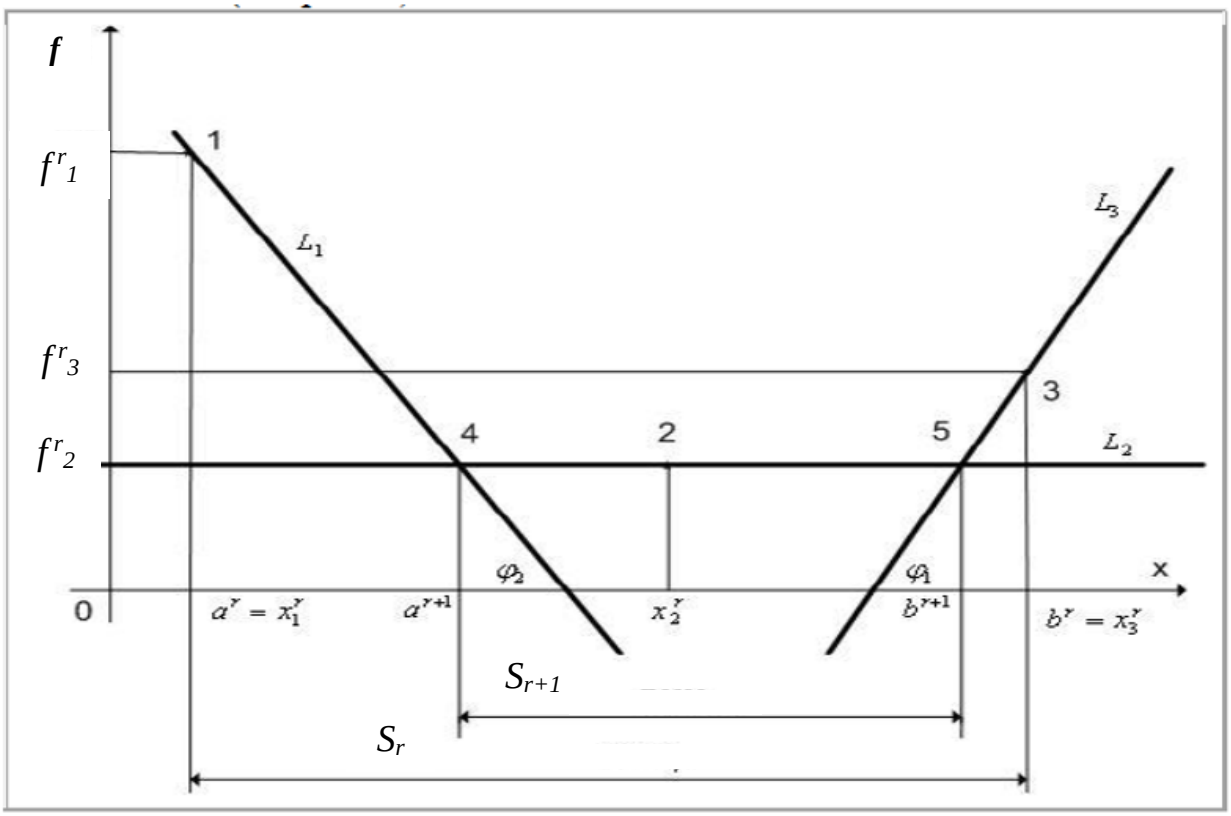


Рис. 2.5.6.1. Звуження інтервалу невизначеності з константою Ліпшица

$tg(\varphi_1) = K, tg(\varphi_2) = -K, \varphi_2 = 180^\circ - \varphi_1$

Проведемо, крім того, через точку (x_2^r, f_2^r) пряму L_2, паралельну до осі *Ox,* до перетину з прямими L_1, L_3 і позначимо абсциси точок перетину a^{r+1}, b^{r+1}.

У зроблених припущеннях, точки a^{r+1}, b^{r+1} можуть бути використані у якості меж інтервалу невизначеності S_r та S_{r+1}. Іншими словами, у зроблених припущеннях, точка x^* мінімуму функції $f(x)$ не може лежати за межами інтервалу $[a^{r+1}, b^{r+1}] \in [a^r, b^r]$

Знайдемо величини a^{r+1}, b^{r+1}.

Пряма L_1 має рівняння $y_1 = Kx + c$, де константа c визначається з умови проходження цієї прямої через точку (x_1^r, f_1^r)

$$f_1^r = -Kx_1^r + c \Longrightarrow c = f_1^r + Kx_1^r$$

Таким чином, $y_1 = Kx + f_1^r + Kx_1^r$.

У точці 4 має місце рівність $(-K)a^{r+1} + f_1^r + Kx_1^r = f_2^r$, з якої слідує що

$$a^{r+1} = x_1^r + \frac{f_1^r - f_2^r}{K}$$

Аналогічно для прямої L_2

$$b^{r+1} = x_3^r + \frac{f_3^r - f_2^r}{K}$$

Питання до розділу 2.

1. Яка функція називається монотонною на проміжку?
2. Яким може бути графік немонотонної функції ?
3. Що означає твердження, що функція має в точці максимум (мінімум)?
4. Які точки називають стаціонарними?
5. Що означає зміна знаку першої похідної з “-“ на “+” , з “+” на “-“ ?
6. Яка функція називається унімодальною?
7. Чи може дискретна функція бути унімодальною?
8. В чому суть правила виключення інтервалів?
9. Які ви знаєте числові методи одновимірної оптимізації?
10. Що таке “золота” пропорція?
11. Дайте означення поняттям число Фібоначчі, ряд Фібоначчі.

12. Коли доцільно використовувати метод Фібоначчі?

13. Коли метод Фібоначчі переходить в метод “золотого” ділення?

14. В чому суть методів поліноміальної апроксимації?

15. Як будують апроксимаційний поліном в методі кубічної апроксимації?

16. Коли в методі Ньютона доцільно використовувати формули чисельного диференціювання?

РОЗДІЛ 3. БЕЗУМОВНА ОПТИМІЗАЦІЯ ФУНКЦІЇ КІЛЬКОХ ЗМІННИХ

3.1. Основні поняття

3.1.1. Умови існування екстремуму

Означення. Множина точок $M(x_1, x_2, ..., x_n)$, координати яких незалежно одна від одної задовольняють нерівностям

$$\begin{aligned} &a_1 \le x_1 \le b_1, \\ &a_2 \le x_1 \le b_2, \\ &\quad \dots \\ &a_n \le x_1 \le b_n \end{aligned} \qquad (3.1.1.1)$$

називається n – вимірним замкнутим “прямокутним паралелепіпедом” і скорочено позначається:

$[a_1, b_1; a_2, b_2; ...; a_n, b_n]$.

Якщо в співвідношеннях (3.1.1.1) відсутній знак “дорівнює”, тобто

$$\begin{aligned} &a_1 < x_1 < b_1 \\ &a_2 < x_1 < b_2 \\ &\quad \dots \\ &a_n < x_1 < b_n \end{aligned}$$

то відповідна область називається *відкритим* “прямокутним паралелепіпедом” і скорочено позначається:

$(a_1, b_1; a_2, b_2; ...; a_n, b_n)$.

Різниці $b_1 - a_1; b_2 - a_2; ...; b_n - a_n$ називають *вимірами* обох паралелепіпедів, а точку $\left(\frac{a_1 + b_1}{2}; \frac{a_2 + b_2}{2}; ...; \frac{a_n + b_n}{2}\right)$ - *центром* паралелепіпедів.

Означення. *Околицею* точки $M_0(x_1^0, x_2^0, ..., x_n^0)$ називається будь-який відкритий “паралелепіпед”

$$\left(x_1^0 - \delta_1, x_1^0 + \delta_1; ...; x_n^0 - \delta_n, x_n^0 + \delta_n\right)$$

$(\delta_1, \delta_2, ..., \delta_n > 0)$ з центром в точці M_0. При однакових δ_i це буде “куб”, або кажуть “гіперкуб”

$$\left(x_1^0 - \delta, x_1^0 + \delta; ...; x_n^0 - \delta, x_n^0 + \delta\right),$$

всі виміри якого дорівнюють 2δ.

Означення. Множина точок $M(x_1, x_2, ..., x_n)$, координати яких задовольняють нерівності

$$x_1 \geq 0,\ x_2 \geq 0, \ldots x_n \geq 0, \qquad x_1 + x_2 + \ldots + x_n \leq h \quad (h > 0)$$

називається замкнутим *симплексом* (якщо в наведених умовах виключити знак рівності, то *симплекс* називають відкритим).

При $n = 2$ геометричним образом цієї множини точок буде рівнобедрений прямокутний трикутник, а при $n = 3$ - тетраедр. В $n-$ вимірному просторі *симплекс* це найпростіша багатогранна область, з найменш можливою для даного простору кількістю граней.

Означення. Множина точок $M(x_1, x_2, ..., x_n)$, координати яких задовольняють нерівності

$$(x_1 - x_1^0)^2 + (x_2 - x_2^0)^2 + ... + (x_n - x_n^0)^2 \leq r^2$$

де $M_0(x_1^0, x_2^0, ..., x_n^0)$ - деяка точка, а r – фіксоване число більше нуля, утворює замкнуту (або відкриту) n – вимірну *сферу* радіуса r з центром в точці M_0.

Означення. *Сферичною околицею* точки $M_0(x_1^0, x_2^0,...,x_n^0)$ називають відкриту “сферу” будь-якого радіуса $r > 0$ з центром у точці M_0.

Зрозуміло, що якщо точку $M_0(x_1^0, x_2^0,...,x_n^0)$ можна оточити околом у вигляді n – вимірного “паралелепіпеда”, то цю ж саму точку можна оточити околом у вигляді n – вимірної “сфери” так, що “сфера” буде вписаною в “паралелепіпед” і навпаки.

Означення. Нехай функція $u = f(x_1, x_2,...,x_n)$ визначена в області D і $M_0(x_1^0, x_2^0,...,x_n^0)$ – внутрішня точка цієї області. Кажуть, що функція $f(x_1, x_2,...,x_n)$ в точці $M_0(x_1^0, x_2^0,...,x_n^0)$ має максимум (мінімум), якщо існує такий окіл точки M_0

$$\left(x_1^0 - \delta, x_1^0 + \delta;...; x_n^0 - \delta, x_n^0 + \delta\right)$$

що для будь-якої точки $M(x_1, x_2,...,x_n)$ цього околу виконується нерівність

$f(x_1 x_2...,x_n) \le f(x_1^0, x_2^0,...,x_n^0)$ для точки *максимуму* (3.1.1.2)

або

$f(x_1 x_2...,x_n) \ge f(x_1^0, x_2^0,...,x_n^0)$ для точки *мінімуму*. (3.1.1.3)

Функція $u = f(x_1, x_2,...,x_n)$, для якої відшукується екстремум, зветься *цільовою* функцією.

Означення. Якщо окіл точки $M_0(x_1^0, x_2^0,...,x_n^0)$ можна вибрати таким, що для будь-якої його точки буде виконуватись строга нерівність виду (3.1.1.2)-(3.1.1.3), то кажуть, що в точці M_0 має місце *власний* максимум (або власний мінімум).

Для позначення максимуму і мінімуму функції кількох змінних використовують загальний термін *екстремум*.

У курсі вищої математики доводять [12], що якщо функція $f(x_1, x_2,...,x_n)$

в деякій точці $M_0(x_1^0, x_2^0, ..., x_n^0)$ має екстремум, то всі частинні похідні першого порядку цієї функції в точці M_0 дорівнюють нулю. При цьому вважається, що функція $f(x_1, x_2, ..., x_n)$ є неперервною і частинні похідні першого порядку в точці M_0 існують.

Отже, екстремум може бути в тих точках, в яких частинні похідні першого порядку всі дорівнюють нулю. Такі точки називають *стаціонарними*. Координати цих точок можна знайти, розв'язавши систему рівнянь

$$\begin{cases} f'_{x_1}(x_1, x_2, ..., x_n) = 0 \\ f'_{x_2}(x_1, x_2, ..., x_n) = 0 \\ ... \\ f'_{x_n}(x_1, x_2, ..., x_n) = 0 \end{cases}.$$

Екстремум може бути в стаціонарній точці. Якщо $M_0(x_1^0, x_2^0, ..., x_n^0)$ стаціонарна точка, то щоб відповісти на запитання, чи є в цій точці екстремум, треба дослідити поведінку приросту функції

$$\Delta = f(x_1, x_2, ..., x_n) - f(x_1^0, x_2^0, ..., x_n^0)$$

в околі стаціонарної точки [8].

Якщо приріст функції Δ в околі стаціонарної точки $M_0(x_1^0, x_2^0, ..., x_n^0)$ розкласти по формулі Тейлора, то одержимо

$$\Delta = \frac{1}{2}\{ f''_{x_1x_1}\Delta x_1^2 + ... + f''_{x_nx_n}\Delta x_n^2 + 2f''_{x_1x_2}\Delta x_1\Delta x_2 + 2f''_{x_1x_3}\Delta x_1\Delta x_3 + ... +$$
$$+ 2f''_{x_1x_n}\Delta x_1\Delta x_n + ... + 2f''_{x_{n-1}x_n}\Delta x_{n-1}\Delta x_n\} = \frac{1}{2}\sum_{i,k}^{n} 2f''_{x_ix_k}\Delta x_i\Delta x_k$$

У цій формулі приріст функції в околі стаціонарної точки виражено через прирости аргументів $\Delta x_i = x_i - x_i^0$.

При цьому, згідно з теорією формули Тейлора [14; 16], всі похідні обчислені в деякій проміжній точці

$$\left(x_1^0 + \theta\Delta x_1; x_2^0 + \theta\Delta x_2; ...; x_n^0 + \theta\Delta x_n\right), \quad (0 < \theta < 1).$$

Якщо ввести позначення

$$f''_{x_i x_k}(x_1^0, x_2^0, ..., x_n^0) = a_{ik}, \quad (i,k = 1,2,...n), \tag{3.1.1.4}$$

то згідно з формулою Тейлора

$$f''_{x_i x_k}(x_1^0 + \theta\Delta x_1, ..., x_n^0 + \theta\Delta x_n) = a_{ik} + \alpha_{ik},$$

причому $\alpha_{ik} \to 0$ при $\Delta x_i \to 0, \quad i = 1,2,...n$.

Тепер вираз для приросту функції Δ в околі стаціонарної точки можна представити у вигляді

$$\Delta = \frac{1}{2}\left\{ \sum_{i,k=1}^{n} a_{ik}\Delta x_i \Delta x_k + \sum_{i,k=1}^{n} \alpha_{ik}\Delta x_i \Delta x_k \right\}. \tag{3.1.1.5}$$

В (3.1.1.5) перша сума є другим диференціалом функції $f(x_1, x_2, ..., x_n)$ в досліджуваній стаціонарній точці $M_0(x_1^0, x_2^0, ..., x_n^0)$, а друга сума прямує до нуля при $\Delta x_i \to 0$. Таким чином, приріст функції в околі стаціонарної точки визначається властивостями другого диференціалу функції в цій точці.

Вираз

$$\sum_{i,k=1}^{n} a_{ik}\Delta x_i \Delta x_k \tag{3.1.1.6}$$

являє собою однорідний многочлен другого ступеню, або квадратичну форму від змінних $\Delta x_1, \Delta x_2, ..., \Delta x_n$. Від властивостей цієї квадратичної форми і залежить поведінка приросту функції в околі стаціонарної точки.

Для побудови виразу другого диференціалу функції кількох змінних треба в фіксованій точці $M_0(x_1^0, x_2^0, ..., x_n^0)$ обчислити всі коефіцієнти a_{ik} по формулі (3.1.1.4). Коефіцієнти a_{ik} є елементами матриці

$$\{a_{ik}\}_{i,k=1}^{n} = \{\frac{\partial^2 f}{\partial x_i \partial x_k}\}_{i,k=1}^{n} = \begin{pmatrix} f''_{x_1 x_1} & f''_{x_1 x_2} & \cdots & f''_{x_1 x_n} \\ f''_{x_2 x_1} & f''_{x_2 x_2} & \cdots & f''_{x_2 x_n} \\ \cdots & \cdots & \cdots & \cdots \\ f''_{x_n x_1} & f''_{x_n x_2} & \cdots & f''_{x_n x_n} \end{pmatrix} \tag{3.1.1.7}$$

Матрицю (3.1.1.7) називають матрицею Гессе, або гессіаном. Вона широко використовується в теорії екстремальних задач. Властивості квадратичної форми як певного математичного об'єкту вивчають в курсі вищої алгебри.

Означення. Квадратичну форму

$$\sum_{i,k=1}^{n} a_{ik} y_i y_k \qquad (3.1.1.8)$$

називають додатньо визначеною (від'ємно визначеною), якщо вона набуває додатних (від'ємних) значень при будь-яких значеннях аргументів y_i, які одночасно не дорівнюють нулю.

Наприклад, наступна форма є додатньо визначеною

$$6y_1^2 + 5y_2^2 + 14y_3^2 + 4y_1y_2 - 8y_1y_3 - 2y_2y_3. \qquad (3.1.1.9)$$

Це стає очевидним, якщо записати її у вигляді

$$(2y_1 - 3y_3)^2 + (y_1 + y_2 + y_3)^2 + (y_2 - y_3)^2.$$

Квадратичну форму (3.1.1.9) можна записати і так

$$\begin{bmatrix} y_1 y_2 y_3 \end{bmatrix} \begin{pmatrix} 6 & 2 & -4 \\ 2 & 5 & -1 \\ -4 & -1 & 14 \end{pmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = (y, Ay). \qquad (3.1.1.10)$$

У виразі (3.1.1.10) матрицю A називають матрицею квадратичної форми. У вищій алгебрі при вивченні властивостей квадратичних форм доводять теорему Сильвестра, в якій сформульовано необхідні й достатні умови для того, щоб квадратична форма (3.1.1.8) була додатньо визначеною [14].

Теорема Сильвестра. Для того, щоб квадратична форма (3.1.1.8) була додатньо визначеною необхідно й достатньо, щоб усі головні мінори матриці квадратичної форми були додатними.

Математично теорема Сильвестра записується у вигляді ланцюжка нерівностей:

$$a_{11} > 0; \quad \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} > 0; \ \ldots . \ \begin{vmatrix} a_{11} & \ldots & a_{1n} \\ \ldots & \ldots & \ldots \\ a_{n1} & \ldots & a_{nn} \end{vmatrix} > 0;$$

Від'ємно визначена квадратична форма стає додатньо визначеною, якщо в матриці квадратичної форми поміняти знаки всіх її елементів на протилежні. Отже, для того, щоб квадратична форма була від'ємно визначеною, необхідно й достатньо, щоб усі головні мінори непарного порядку були від'ємними, а головні мінори парного порядку – додатніми.

Використовуючи введені поняття, сформулюємо достатні умови для існування екстремуму функції кількох змінних [11].

Якщо другий диференціал, тобто квадратична форма

$$\sum_{i,k=1}^{n} a_{ik} \Delta x_i \Delta x_k \qquad (3.1.1.11)$$

зі значеннями коефіцієнтів a_{ik}, обчисленими за формулою (3.1.1.4), є додатньо визначеною квадратичною формою, то в досліджуваній стаціонарній $M_0(x_1^0, x_2^0, \ldots, x_n^0)$ буде власний мінімум. Якщо ж ця форма буде від'ємно визначеною, то в досліджуваній стаціонарній точці $M_0(x_1^0, x_2^0, \ldots, x_n^0)$ буде власний максимум.

Квадратична форма (3.1.1.8) називається невизначеною, якщо вона може набувати значення різних знаків. Такою, наприклад, є форма

$6y_1^2 + y_2^2 + y_3^2 + 8y_1y_2 - 8y_1y_3 - 2y_2y_3$.

Дійсно, якщо покласти $\bar{y} = \{1;0;0\}$, то

$$\begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{pmatrix} 6 & 4 & -4 \\ 4 & 1 & -1 \\ -4 & -1 & 1 \end{pmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = +6.$$

Якщо ж покласти $\bar{y} = \{1;-1;0\}$, то

$$\begin{bmatrix}1 & -1 & 0\end{bmatrix}\begin{pmatrix}6 & 4 & -4\\ 4 & 1 & -1\\ -4 & -1 & 1\end{pmatrix}\begin{bmatrix}1\\ -1\\ 0\end{bmatrix}=-1.$$

Висновок. Якщо квадратична форма (3.1.1.11) буде невизначеною, то в досліджуваній стаціонарній точці $M_0(x_1^0, x_2^0, ..., x_n^0)$ екстремум відсутній. Ознакою відсутності екстремуму в даному прикладі є невиконання критерію Сильвестра.

$$6>0; \qquad \begin{vmatrix}6 & 4\\ 4 & 1\end{vmatrix}=-10; \qquad \begin{vmatrix}6 & 4 & -4\\ 4 & 1 & -1\\ -4 & -1 & 1\end{vmatrix}=0.$$

Можливі випадки, коли квадратична форма не може набувати значення різних знаків, але не є визначеною, бо перетворюється в нуль не лише при нульових значеннях аргументів. Таку форму називають напіввизначеною. Такою, наприклад, є форма

$$y_{12}+y_{22}+y_{32}+2y_1y_2+2y_1y_3+2y_2y_3=(y_1+y_2+y_3)^2.$$

Ця форма не може набувати від'ємні значення, але ця форма дорівнює нулю при $y_1+y_2+y_3=0$.

Випадок, коли квадратична форма є напіввизначеною, є "сумнівним" випадком. Залежно від поведінки вищих похідних екстремум або є, або його немає [12].

Нехай A - дійсна симетрична матриця, а T - ортогональна матриця така, що виконується співвідношення

$T^TAT=\Lambda$, де Λ - діагональна матриця.

Введемо нову змінну $\bar{y}=T^T\bar{x}$ і квадратичну форму $(\bar{x}, A\bar{x})$ у вигляді суми квадратів [2; 11]

$$(\bar{x}, A\bar{x})=\bar{y}^TT^TAT\bar{y}=(\bar{y}^T, \Lambda\bar{y})=\sum_{i=1}^{n}\lambda_i y_i^2.$$

Такий запис дозволяє легко встановити визначеність квадратичної форми.

Квадратична форма додатньо визначена тоді і тільки тоді, коли всі власні значення матриці A додатні [1].

Квадратична форма буде додатньо напіввизначеною тоді і тільки тоді, коли всі власні значення матриці A додатні і серед них є нулі.

Квадратична форма невизначена тоді і тільки тоді, коли серед власних значень матриці A є і додатні і від'ємні.

3.1.2. Рельєф функції

Основні проблеми багатовимірного випадку знаходження екстремуму зручно розглянути на прикладі функції двох змінних [2; 10].

Функції $z = F(x, y)$ в тривимірному просторі відповідає деяка поверхня. Рельєф цієї поверхні можна зобразити лініями рівня так, як це прийнято в топографії. Зафіксуємо ряд площин $z = C_i$ і знайдемо лінії їх перетину з поверхнею $F(x, y)$. Проекції цих ліній на площину XOY називають лініями рівня (рис. 3.1.2.1). Сукупність ліній рівня дає уявлення про рельєф функції $z = F(x, y)$.

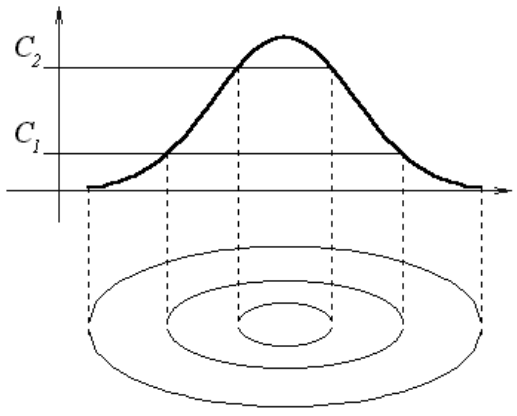


Рис. 3.1.2.1. Лінії рівня функції $z = F(x, y)$

По виду лінії рівня умовно виділяють три типи рельєфу: котловинний, ярів, невпорядкований [10]. При котловинному рельєфі лінії рівня схожі на еліпси (рис. 3.1.2.2).

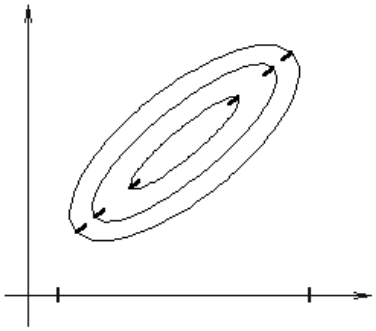


Рис. 3.1.2.2. Котловинний тип рельєфу функції

В малому околі невиродженого мінімуму (максимуму) рельєф функції завжди котловинний. Дійсно, в точці екстремуму гладкої функції виконуються необхідні умови існування екстремуму:

$$\frac{\partial F}{\partial x} = \frac{\partial F}{\partial y} = 0.$$

Якщо функцію кількох змінних розкласти по формулі Тейлора в точці $\bar{x}^0$, то матимемо:

$$f(\bar{x}) = f(\bar{x}^0) + \nabla f(\bar{x}^0)^T \Delta\bar{x}^0 + \frac{1}{2}\Delta\bar{x}^{0T}\nabla^2 f(\bar{x}^0)\Delta\bar{x}^0 + O_3(\Delta\bar{x}) \qquad (3.1.2.1)$$

де $\Delta\bar{x}$ – вектор приростів аргументів $\Delta\bar{x}^0 = \bar{x} - \bar{x}^0 = \{x_i - x_i^0\}_{i=1}^N$;

$\nabla f(\bar{x}^0)$ – N–вимірний вектор-стовпчик перших похідних $f(\bar{x})$, обчислених в точці $\bar{x}^0$; $\nabla^2 f(\bar{x}^0)$ – симетрична матриця $N \times N$ других частинних похідних, або матриця Гессе.

Якщо $\bar{x}^0$ стаціонарна точка, то розклад (3.1.2.1) для функції двох змінних матиме вигляд

$$F(x,y) = F(x_0,y_0) + \frac{1}{2}(\Delta x)^2 F''_{xx} + \Delta x \Delta y F''_{xy} + \frac{1}{2}(\Delta y)^2 F''_{yy} + \ldots \qquad (3.1.2.2)$$

Якщо в точці $\bar{x}^0$ має місце екстремум, то квадратична форма (3.1.2.2) буде визначеною додатньо або від'ємно. Лініями рівня знаковизначеної квадратичної форми є еліпси.

Розглянемо тип рельєфу яр (рис. 3.1.2.3). Якщо лінії рівня кусочно гладкі,

то виділимо на кожній з них точку зламу.

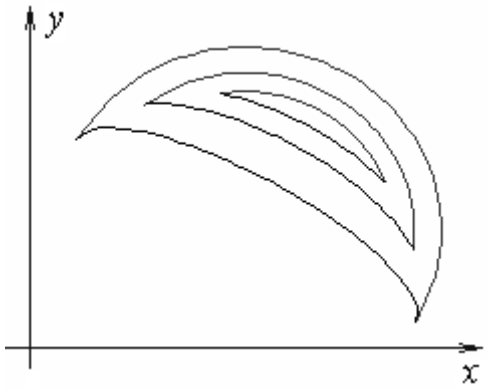


Рис. 3.1.2.3. Тип рельєфу яр з точками зламу

Означення. Геометричне місце точок зламу називають *істинним яром*, якщо кут зламу направлений в сторону зростання функції.

Означення. Геометричне місце точок зламу називають *істинним гребнем*, якщо кут зламу направлений в сторону зменшення функції.

В практичних задачах частіше буває, що лінії рівня скрізь гладкі, але на них є ділянки з великою кривизною (рис. 3.1.2.4).

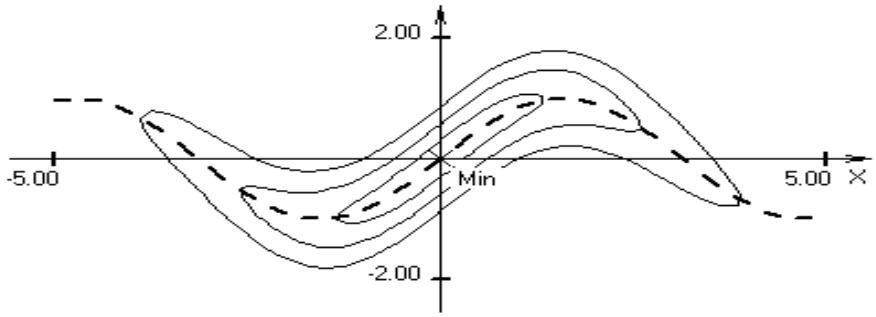


Рис. 3.1.2.4. Тип рельєфу яр з гладкими лініями рівня

Геометричні місця точок з великою кривизною називають *розв'язуваними* ярами або гребнями. Наприклад, рельєф функції $F(x,y)=10(y-\sin x)^2+0{,}1x^2$, зображений на рис. 3.1.2.4., має яскраво виражений звивистий розв'язуваний яр, "дно" якого – синусоїда, а найнижча точка – початок координат. В фізичних задачах тип рельєфу яр вказує на те, що при формулюванні задачі не враховано якийсь зв'язок між певними змінними.

Невпорядкований тип рельєфу характеризується наявністю багатьох

максимумів, мінімумів і сідловин. Прикладом може бути функція

$$F(x,y)=(1+\sin^2 x)(1+\sin^2 y),$$

яка має мінімуми в точках $x_k=\pi k;\ \ y_j=\pi j$ і максимуми в точках, які **здвигнуті** відносно мінімумів на $\pi/2$ по кожній координаті.

3.1.3. Характеристика методів

Всі ефективні методи знаходження екстремумів зводяться до побудови траєкторій, вздовж яких цільова функція зменшується (зростає) відповідно при пошуку мінімуму (або максимуму) [2; 3; 11]. Різні методи відрізняються способами побудови таких траєкторій. Метод, пристосований до одного типу рельєфу, може виявитись неефективним на рельєфах іншого типу. Методи, орієнтовані на розв'язування задач безумовної оптимізації, можна умовно поділити на три широкі класи в залежності від використовуваної в алгоритмі інформації:

1. Методи прямого пошуку (методи нульового порядку), в алгоритмах яких обчислюються лише значення цільової функції.

2. Градієнтні методи (методи першого порядку), в алгоритмах яких використовуються значення перших частинних похідних цільової функції.

3. Методи другого порядку, в алгоритмах яких разом з першими частинними похідними використовуються другі частинні похідні цільової функції.

Жоден із існуючих на сьогодні методів оптимізації не є оптимальним. На практиці при розв'язуванні екстремальних задач часто виникають ситуації коли:

- доводиться працювати з дуже великими і дуже малими числами;
- обчислення значень цільової функції коштує дуже дорого.

Спроби вийти з такої ситуації, в свою чергу, вимагають надмірних затрат машинного часу.

Є задачі в яких або неможливо, або дуже важко знайти аналітичні вирази

для похідних цільової функції. При застосуванні градієнтних методів доводиться користуватись різницевою апроксимацією похідних. Це вимагає експериментальної перевірки величин кроків диференціювання і спусків, які забезпечують відповідність похибки заокруглення і похибки апроксимації похідних.

Питання до підрозділу 3.1.

1. Як слід розуміти термін окіл в n-вимірному просторі?
2. Що таке замкнутий симплекс в n-вимірному просторі?
3. Коли функція n-змінних має в в точці мінімум (максимум)?
4. Сформулюйте необхідні умови існування екстремуму функції n-змінних.
5. Що таке друга похідна?
6. Дайте означення квадратичної форми.
7. Якою може бути квадратична форма?
8. Сформулюйте критерій Сильвестра.
9. Дайте означення матриці Гессе.
10. Коли квадратична форма буде невизначеною?
11. Що таке поверхня рівня?
12. Які види рельєфу ви знаєте?
13. Що означають терміни яр, гребінь, западина?
14. Чим відрізняються евристичні і теоретичні методи?

3.2. Методи прямого пошуку

3.2.1. Метод пошуку по симплексу

Методи безумовної оптимізації, в алгоритмах яких використовується лише обчислення значень цільової функції, можна поділити на евристичні і теоретичні. Евристичні методи реалізують процедуру пошуку екстремуму на основі інтуїтивних геометричних уявлень. Теоретичні методи основані на

використанні досягнень і апарату сучасного математичного аналізу. В таких методах завжди докладно аналізується швидкість їх збіжності [2; 3; 11].

Якщо треба знайти екстремум функції однієї змінної, то перше уявлення про його місцезнаходження дає таблиця значень цієї функції на досліджуваному відрізку зміни аргументу. Аналогічно, якщо треба знайти екстремум функції двох змінних, то уявлення про його місцезнаходження дає двовимірна таблиця значень функції, побудована для досліджуваної області.

В евристичних методах область неперервної зміни аргументів цільової функції замінюється дискретною множиною (сіткою) точок простору, а потім використовуються різні стратегії зменшення області, яка містить розв'язок. При цьому, якщо алгоритм методу зводиться до перегляду значень функції в усіх точках сітки, то такий метод неефективний для функцій, у яких кількість змінних більша ніж дві.

Можна поступити інакше. Вибираємо деяку “базову“ точку і обчислюємо в ній значення функції. Потім оцінюємо значення цільової функції в точках, які оточують “базову” точку. Наприклад, при розв’язуванні задачі з двома змінними можна використати шаблон з п’яти точок (рис. 3.2.1.1).

Рис. 3.2.1.1. Шаблон з п’яти точок

“Найкраща“ із п’яти точок приймається за “базову” і виконується наступний крок методу. Якщо жодна з оточуючих точок “не є кращою” за “базову”, то розмір шаблону зменшують і продовжують пошук [5; 11].

У багатовимірних задачах обчислення значень цільової функції виконують в усіх вершинах і в центрі “гіперкуба”. Якщо кількість змінних

дорівнює N, то пошук по одному “гіперкубу” вимагає виконання $2^N + 1$ обчислень значень функції і метод стає неефективним для великих значень N.

Означення. Регулярним симплексом в N – вимірному просторі називають многогранник, який має $N+1$ вершину і вершини якого є рівновіддаленими точками.

В двовимірному просторі симплексом є рівносторонній трикутник. В тривимірному просторі симплексом є тетраедр.

В алгоритмі пошуку по симплексу (S^2-метод) використовують важливу властивість симплексів: новий симплекс можна побудувати на будь-якій грані вихідного симплекса шляхом переносу вибраної вершини на певну відстань вздовж прямої, проведеної через центр ваги решти вершин симплекса (рис. 3.6).

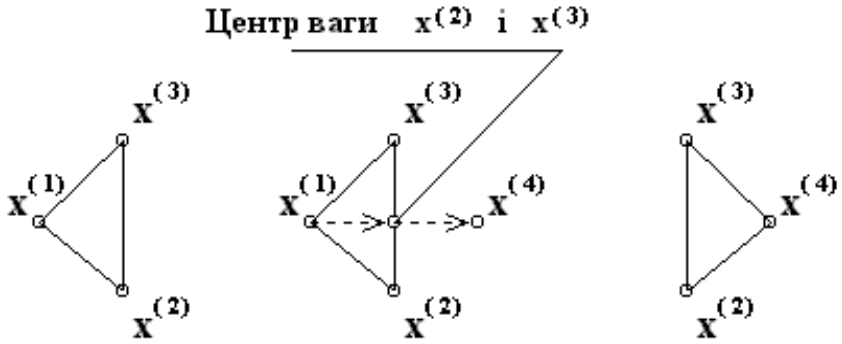


Рис. 3.6. Процес побудови нового симплекса на площині

Одержана таким чином точка буде вершиною нового симплекса. Вершина, яку використали при побудові, виключається з розгляду.

Якщо задана базова точка $\bar{x}^{(0)}$, то в N – вимірному просторі координати решти N вершин симплекса обчислюються по формулі

$$\bar{x}^{(i)} = \begin{cases} x_j^{(0)} + \delta_1, & j \neq 1, \\ x_j^{(0)} + \delta_2, & j = 1, \end{cases} \quad \text{для } i, j = 1,2,...N .$$

Прирости δ_1 і δ_2 залежать лише від N і вибраного масштабного

множника α і обчислюються по формулах

$$\delta_1 = \left[\frac{\sqrt{N+1} + N - 1}{N\sqrt{2}} \right] \alpha, \qquad \delta_2 = \left[\frac{\sqrt{N+1} - 1}{N\sqrt{2}} \right] \alpha .$$

Величину масштабного множника α вибирають, виходячи із характеристик задачі. Якщо $\alpha = 1$, то ребра регулярного симплексу мають одиничну довжину.

Якщо $\bar{x}^{(j)}$ – вершина, яка проектується, то центр ваги решти *N* вершин обчислюється по формулі

$$\bar{x}^{(c)} = \frac{1}{N} \sum_{i=0, i \neq j}^{N} \bar{x}^{(i)} .$$

Рівняння прямої, яка проходить через точки $\bar{x}^{(j)}$ і $\bar{x}^{(c)}$ має вигляд:

$$\bar{x} = \bar{x}^{(j)} + \lambda (\bar{x}^{(c)} - \bar{x}^{(j)}).$$

При $\lambda = 0$ одержимо вихідну точку $\bar{x}^{(j)}$; якщо $\lambda = 1$, то попадемо в центр ваги, тобто точку $\bar{x}^{(c)}$. Для того, щоб побудований симплекс був регулярним, відображення повинно бути симетричним, і тому нову вершину одержимо, поклавши $\lambda = 2$.

Алгоритм методу пошуку по симплексу буде таким [11; 13; 15]:

1) будуємо регулярний симплекс в просторі незалежних змінних і обчислюємо значення цільової функції в кожній з його вершин;

2) знаходимо вершину з найбільшим (найменшим) значенням цільової функції (“найкращу” вершину);

3) знаходимо центр ваги вершин симплекса, без урахування “найкращої” вершини;

4) проектуємо “найкращу” вершину через центр ваги решти вершин в нову точку, яка буде вершиною нового симплекса;

5) повторюємо цикл, перейшовши на пункт 1.

Якщо цільова функція змінюється плавно (є гладкою), то в ході виконання ітераційного процесу можливі два випадки:

1. Черговий симплекс “накриває” точку екстремуму. Якщо вершина, якій відповідає “найкраще” значення цільової функції побудована на попередній ітерації, то замість неї беруть вершину, якій відповідає наступне за величиною значення цільової функції.

2. Якщо деяка вершина симплексу не зникає на протязі виконання більш ніж M ітерацій, то треба зменшити розміри симплексу з допомогою коефіцієнта редукції і побудувати новий симплекс, приймаючи за базову точку, в якій значення цільової функції “найкраще”. Рекомендується обчислювати M по емпіричній формулі: $M = 1{,}65N + 0{,}05N^2$, де N – кількість змінних задачі, і заокруглювати M до найближчого цілого.

Ітераційний процес пошуку по симплексу закінчується, коли розміри симплексу стають малими, або значення функції в вершинах симплексу мало відрізняються.

Розглянутий алгоритм пошуку по симплексу має ряд переваг, найважливішими з яких є:

– логічна схема методу нескладна. Метод легко програмується;

– для реалізації методу достатньо мати в програмі масив чисел розмірністю $(N+1, N+2)$;

– метод можна пристосовувати до конкретної задачі шляхом зміни (підбору) кількох параметрів, а саме: масштабного множника α, коефіцієнта зміни множника α, параметрів закінчення пошуку.

Недоліком методу є те, що він досить повільний, оскільки інформація, одержана на попередніх ітераціях, ніяк не використовується для прискорення пошуку екстремуму. Можливе наступне удосконалення розглянутого S^2-методу, при якому згладжуються деякі недоліки.

3.2.2. Метод Нелдера–Міда

Зрозуміло, що в процесі пошуку екстремуму S2-методом регулярність симплекса не є обов’язковою. Отже, при побудові нового симплекса, можна

його або розтягнути, або стиснути.

В симплексному алгоритмі Нелдера і Міда [15] мінімізація функції n змінних, здійснюється за допомогою використання деформованого багатогранника.

Будемо розглядати k-ю ітерацію алгоритму. Шлях $\overline{\bar{x}_i^k} = \left[x_{t1}^k, x_{t2}^k, \dots, x_{tn}^k\right]^T, i = 1, \dots, (n+1)$, є i – та вершина в E^n на k – ом етапі пошуку, k = 0,1,2,…, і нехай значення кінцевої функції в цій вершині $f(\bar{x}_t^k)$. Варто відзначити вершини з мінімальним і максимальним значеннями. І позначимо їх наступним чином:

$f(\bar{x}_h^k) = \max\{f(\bar{x}_1^k), \dots, f(\bar{x}_{n+1}^k)\}$;

$f(\bar{x}_i^k) = \min\{f(\bar{x}_1^k), \dots, f(\bar{x}_{n+1}^k)\}$.

Багатогранник в E^n складається з n+1 вершин $\bar{x}_1^k, \bar{x}_2^k, \dots, \bar{x}_{n+1}^k$. Позначимо через $\bar{x}_{n+2}^k$ - центр ваги вершин без точки $\bar{x}_h^k$ з максимальним значенням функції. Координати цього центру рахуємо за формулою:

$x_{n+2,j}^k = \frac{1}{n}\left[\sum_{i=1}^{n+1} x_{i,j}^k - x_{h,j}^k\right], j = 1, \dots, n.$

Початковий багатогранник зазвичай обирається у вигляді постійного симплекса (з вершиною на початку координат). Можна початок координат помістити в центр ваги. Процедура пошуку вершин в E^n, в яких $f(\bar{x})$ має краще значення, складається з наступних операцій: 1) відображення; 2) розтягнення; 3) стискання; 4) редукція.

1. ***Відображення.*** Відображення являє собою проектовані точки $\bar{x}_h^k$ через центр ваги $\bar{x}_{n+2}^k$ у відповідності з наступним співвідношенням: $\bar{x}_{n+3}^k = \bar{x}_{n+2}^k + \alpha \cdot \left(\bar{x}_{n+2}^k - \bar{x}_h^k\right)$, де α > 0 – коефіцієнт відображення.

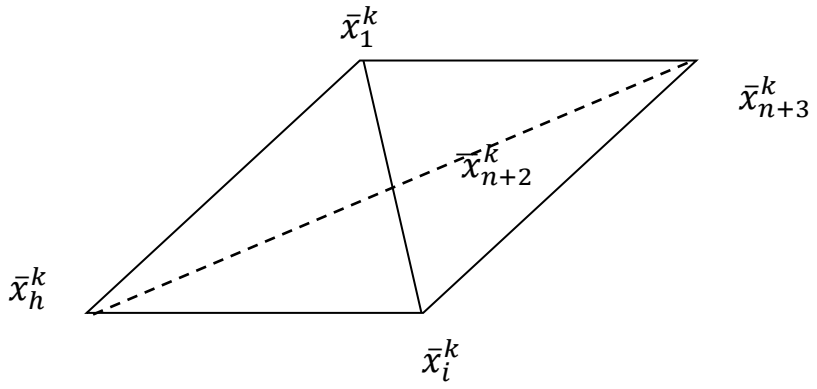


Рис. 3.2.2.1

Порахуємо значення функції в знайденій точці $f(\bar{x}_{n+3}^k)$. Якщо значення функції в даній точці $f\left(\bar{x}_{n+3}^k\right) \geq f\left(\bar{x}_h^k\right)$, то переходимо до четвертого пункту алгоритму – операції редукції для поточного симплексу.

Якщо $f\left(\bar{x}_{n+3}^k\right) < f(\bar{x}_h^k) \wedge f\left(\bar{x}_{n+3}^k\right) < f\left(\bar{\bar{x}}_1^k\right)$, то виконуємо операцію розтягнення для відображеного симплексу.

В протилежному випадку, якщо $f\left(\bar{x}_{n+3}^k\right) < f(\bar{x}_h^k) \wedge f\left(\bar{x}_{n+3}^k\right) \geq f\left(\bar{x}_i^k\right)$, то виконується операція стискання для відображеного симплексу.

2. Розтягнення. Ця операція полягає в наступному. Якщо $f\left(\bar{x}_{n+3}^k\right) < f(\bar{x}_i^k)$ (менше мінімального значення на k-м етапі), то вектор $\left(\bar{x}_{n+3}^k\right) - (\bar{x}_{n+2}^k)$ розтягнеться в відношенні з співвідношенням

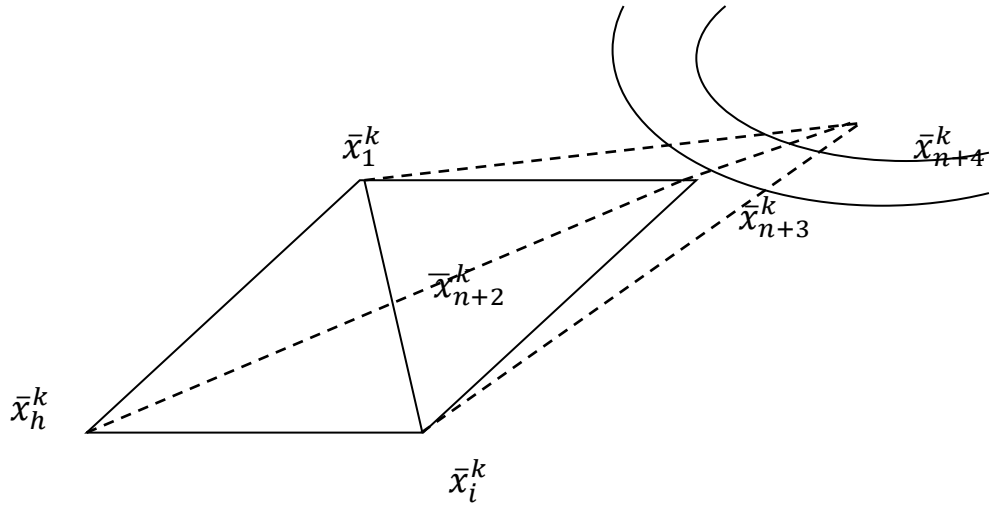


Рис. 3.2.2.1.

$$\bar{x}_{n+3}^k = \bar{x}_{n+2}^k + \gamma \cdot \left(\bar{x}_{n+3}^k - \bar{x}_{n+2}^k\right),$$

де $\gamma > 0$ – коефіцієнт розтягнення.

Якщо $f(\bar{x}_{n+4}^k) < f\left(\bar{x}_l^k\right)$, то точкою $\bar{x}_l^k$ в новому симплексі стає точка $\bar{x}_{n+4}^k$ і процедура продовжується з операції відображення при k = k +1. В протилежному випадку точкою $\bar{x}_l^k$ в новому симплексі стає точка $\bar{x}_{n+3}^k$, а також переходимо до операції відображення.

3. Стиснення. Якщо $f\left(\bar{x}_{n+3}^k\right) > f\left(\bar{x}_i^k\right), \forall\, i, i \neq h$, то вектор $(\bar{x}_h^k - \bar{x}_{n+2}^k)$ стискається у відповідність з формулою

$$\bar{x}_{n+5}^k = \bar{x}_{n+2}^k + \beta \cdot \left(\bar{x}_h^k - \bar{x}_{n+2}^k\right),$$

де $0 < \beta < 1$ – коефіцієнт стиснення. Після цього, точка $\bar{x}_h^k$ в симплексі замінюється на $\bar{x}_{n+5}^k$. Далі переходимо до операції відображення з k = k + 1. При цьому заново обирається точка $\bar{x}_h^{k+1}$.

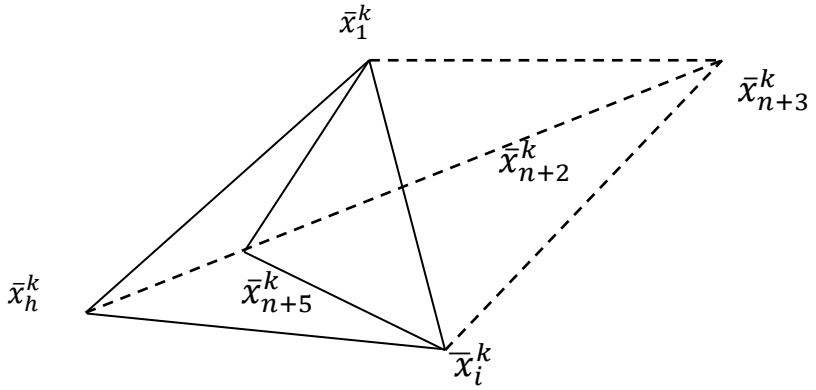


Рис. 3.2.2.2

4. Редукція. Якщо $f\left(\bar{x}_{n+3}^k\right) > f(\bar{x}_h^k)$, то всі вектори $(\bar{x}_i^k - \bar{x}_i^k)$, де i = $\overline{1, (n+1)}$ зменшується в два рази з відліком від точки $\bar{x}_i^k$ за формулою

$$\bar{x}_i^k = \bar{x}_i^k + 0.5 \cdot \left(\bar{x}_i^k - \bar{x}_i^k\right), i = \overline{1, (n+1)}$$

і відбувається перехід до операції відображення (на початок алгоритму з k = k + 1).

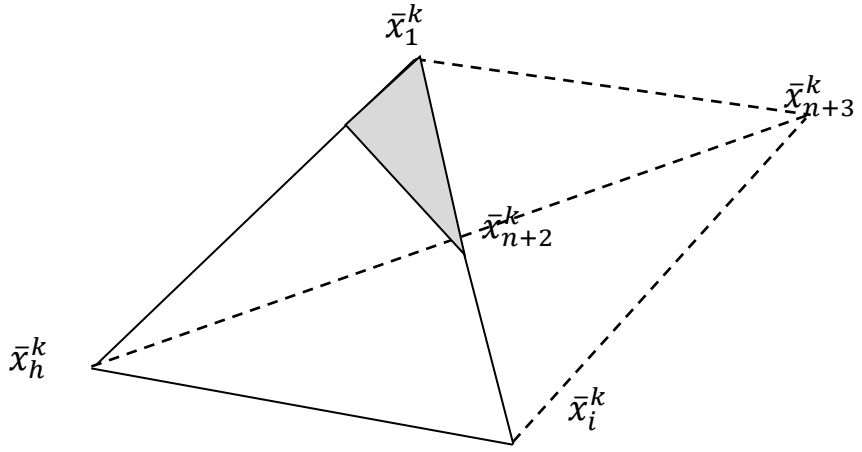


Рис. 3.2.2.3

В якості основного критерія зупинки можуть бути взяті ті самі правила, що і в інших алгоритмах. Можна також використовувати критерій закінчення обчислень

$$\left\{\frac{1}{n+1}\cdot\sum_{i=1}^{n+1}\left[f\left(\bar{x}_{n+2}^k\right)-f\left(\bar{x}_{n+2}^k\right)\right]^2\right\}^{1/2}<\varepsilon.$$

Вибір коефіцієнтів α, β, γ зазвичай здійснюється емпірично. Після того, як многокутник відповідним чином обраний, його розміри повинні підтримуватися незмінними поки зміни в топології задачі не потребують многокутника іншої форми. Частіше за все рекомендують α = 1, 04 ≤ β ≤ 0.6, 2 ≤ γ ≤ 3.

3.2.3. Спуск по координатах

Викладемо алгоритм методу [11; 12] на прикладі пошуку мінімуму функції двох змінних $F(x_1,x_2)$. Виберемо початкове наближення – точку $A(x_{10},x_{20})$.

1) Фіксуємо значення координат, поклавши $x_{2=}\ x_{20.}$ Тоді функція $F(x_1,x_2)$ буде залежати лише від однієї змінної x_1:

$f(x_1)=F(x_1, x_{20})$.

Тепер одним із відомих методів одновимірної оптимізації знайдемо мінімум функції $f(x)$ і позначимо його x_{11}. Тим самим виконано крок з точки $A_0(x_{10},x_{20})$ в точку $B(x_{11},x_{20})$ у напрямку, паралельному координатній осі X_1, і

на цьому кроці значення функції зменшилося.

2) Із одержаної точки B(x_{11},x_{20}) виконаємо крок мінімізації в напрямку, паралельному координатній осі X_2. Для цього розглядаємо функцію одного аргументу:

$f(x2)=F(x_{11},x_2)$,

знаходимо її мінімум по x_2 і позначимо його через x_{21}. Тим самим виконано переходи з точки B(x_{11},x_{20}) в точку C(x_{11},x_{21}).

Перехід в точку C(x_{11},x_{21}) завершує цикл спусків по координатних висях. При повторюванні циклів на кожному спуску функція не збільшується і при цьому значення функції обмежені знизу її мінімумом, а це означає, що ітерації повинні збігатися до деякої межі (рис. 3.2.3.1).

Виникає запитання, чи зійдуться ітерації до мінімуму функції і з якою швидкістю. Це залежить від функції (від її рельєфу) і від вибору початкового наближення. На прикладі функції двох змінних легко переконатися, що можуть бути випадки збіжності методу покоординатного спуску до мінімуму функції і випадки, коли такий спуск до мінімуму не приводить.

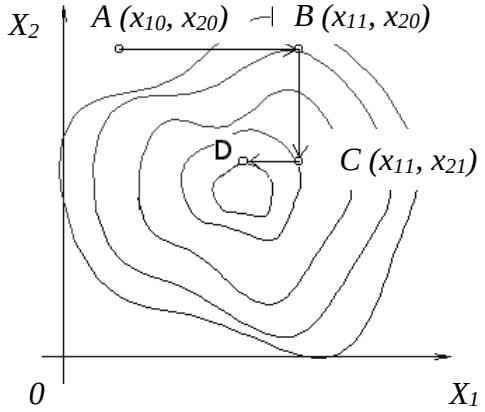


Рис. 3.2.3.1. Геометрична інтерпретація спуску по координатах

Коли рухаємося по вибраному напрямку, тобто по деякій прямій в площині X_1OX_2, то траєкторія руху перетинає лінії рівня і значення функції зменшується або збільшується, залежно від напрямку руху. Лише в тій точці, де напрямок руху стає дотичним до лінії рівня, функція має екстремум вздовж даного напрямку. Визначивши таку точку, завершуємо в ній спуск по даному

напрямку і повинні вибрати наступний напрямок спуску. Через те, що напрямки вибираємо паралельно координатним осям**,** другий напрямок буде перпендикулярним до першого.

Нехай лінії рівня утворюють істинний яр. Тоді можливий випадок, коли спуск по одній координаті приводить на “дно” яру, а будь-який рух по наступній координаті збільшує значення функції. При цьому спуск по координаті стає неможливим і процес спуску по координатах не збігається до мінімуму функції.

Припустимо, що функція є достатньо гладкою. Це означає, що функція має неперервні другі похідні і її мінімум не вироджений. Виберемо деяку точку (x_{10},x_{20}) і проведемо через неї лінію рівня. Нехай в області, обмеженій цією лінією рівня, виконуються нерівності, які означають додатну визначеність квадратичної форми:

$F``_{x1x1} \geq a>0, \quad F``_{x2x2} \geq b>0, \quad |F``_{x1x2}| \leq c, ab>c^2.$

Тоді можна довести [15], що спуск по координатах з точки (x_{10},x_{20}) збігається до мінімуму функції і швидкість збіжності лінійна.

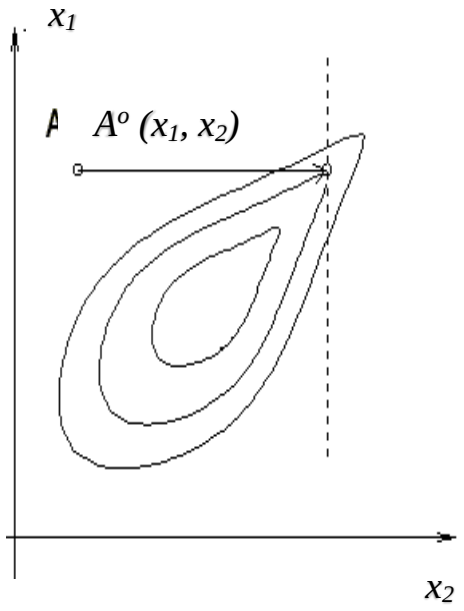


Рис. 3.2.3.2. Приклад відсутності збіжності

Для функції двох змінних швидкість збіжності буде прийнятною, коли

лінії рівня будуть близькими до еліпсів, осі яких паралельні висям координат. Для сильно витягнутих еліпсів, осі яких не співпадають з осями координат, збіжність буде повільною. Якщо збіжність повільна, але траєкторія спуску вже попала в окіл мінімуму, то одержаний результат можна уточнити процесом Ейткена. Зрозуміло, що за вихідні дані при цьому треба брати значення не на трьох послідовних спусках, а на трьох останніх циклах спусків. Розв'язуваний яр можна розуміти як сильно витягнуту котловину. При попаданні траєкторії спуску в такий яр, збіжність стає наскільки повільною, що неможливо продовжувати розв'язування задачі.

Метод спуску по координатах нескладний і легко програмується. Але сходиться він повільно, а при наявності ярів – дуже погано. Тому його використовують як першу спробу при знаходженні мінімуму.

Алгоритм цикличного покоординатного спуску

Початковий етап. Вибрати $eps > 0$, яке буде використовуватися для зупинки алгоритму, і взяти в якості $d_1, ..., d_n$ координатні напрямки. Вибрати початкову точку x_1, покласти $y_1 = x_1$, $k = 1$ та $j = 1$ і перейти до основного етапу.

Основний етап. Крок 1. Покласти lym_j рівним оптимальному вирішенню завдання мінімізації

$f(y_j + lym_j * d_j)$

$lym_j = arg\ min\ f(y_j + lym_j * d_j)$

при умов

$d_i = 1$ при $i=j$ *при цьому в напрямку осі і значення х змінні,*

$d_i = 0$ при $i \neq j$ *при цьому в напрямку осі і значення х не змінні,*

за умови, що lym_j належить *E1(пряма лінія – простір розмірності 1)*.

Покласти $y_{j+1} = y_j + lym_j * d_j$.

Якщо $j < n$, то замінити j на $j + 1$ і повернутися до Кроку 1. Якщо $j = n$, то перейти до Кроку 2.

Крок 2. Покласти $x_{k+1} = y_{n+1}$, якщо $| x_{k+1} - x_k | < eps$, то зупинитися.

В іншому випадку покласти $y_1 = x_{k+1}, j = 1$, замінити k на $k + 1$ і перейти до *Кроку 1*.

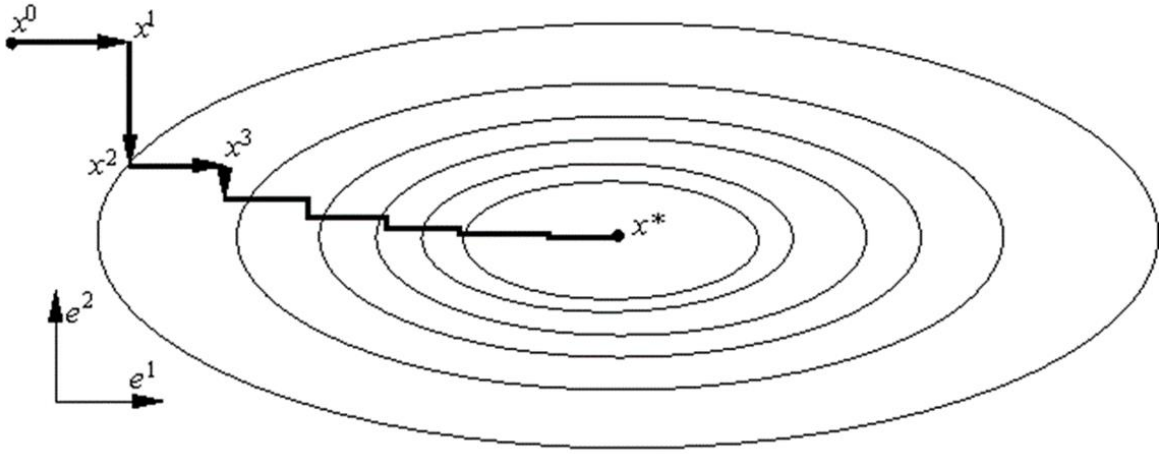
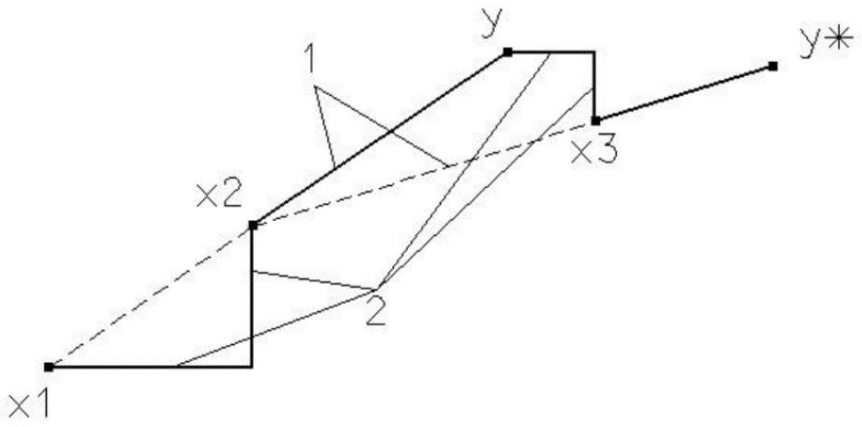


Рис. 3.2.3.3. Циклічний покоординатний спуск.

3.2.4. Метод Хука-Дживса

Метод Хука-Дживса створений в 1961 році і є одним з перших методів у якому при побудові нового напрямку спуску враховується інформація, одержана на попередніх ітераціях [12; 15]. Метод Хука-Дживса здійснює два типи пошуку – дослідницький пошук і пошук за зразком. Перші дві ітерації процедури показані на рисунку.



1 – пошук за зразком

2 – дослідницький пошук вздовж координатних осей

Рис. 3.2.4.1. Кроки методу Хука-Дживса

Послідовність обчислень:

1. При заданому початковому векторі x_1 дослідницький пошук по координатних напрямках приводить в точку x_2.

2. Наступний пошук за зразком в напрямку x_1-x_2 приводить в точку y.

3. Потім дослідницький пошук, який починався з точки y, дає точку x_3.

4. Наступний етап пошуку за зразком вздовж направлення x_3-x_2 дає y^*.

5. Після цього процес повторюється.

Алгоритм Хука-Дживса з використанням одновимірної мінімізації

Розглянемо варіант методу, який використовує одновимірну мінімізіцію вздовж координатних напрямків d_1……..d_2 і направлений пошук за зразком.

Початковий етап. Обрати число eps>0 для зупинки алгоритма. Обрати початкову точку x_1, присвоїти $y_1=x_1$, $k=j=1$ і перейти до основного етапу.

Основний етап.

Крок 1.

Обчислити lym_j – оптимальне рішення задачі мінімізації $f(y_j+lym*d_j)$ за умовою lym належить **E1**. Присвоїти $y_{j+1}=y_j+lym_j* d_j$

– Якщо $j<n$, то замінити j на $j+1$ і повернутись на крок 1.

– Якщо $j=n$, то присвоїти $x_{k+1}=y_{n+1}$

– Якщо $|x_{k+1}-x_k|<eps$, то зупинитися; в протилежному випадку перейти на ***крок 2.***

Крок 2.

Присвоїти $D=x_{k+1} - x_k$ і знайти lym – оптимальний рішення задачі мінімізації $f(x_{k+1}+lym*d_j)$ за умовою lym належить $E1$.

Присвоїти $y_j=x_{k+1}lym*d$, $j=1$, замінити **k** на **k**+1 і перейти на **крок 1**.

3.2.5. Метод Розенброка

Цей ітераційний метод має деяку схожість з алгоритмом Хука і Дживса. Метод Розенрока [2; 3; 13] називають також методом обертових координат. Цей метод помітно ефективніше попередніх методів, особливо при мінімізації функцій типу яр.

Загальна ідея методу полягає в тому, що вибирається система ортогональних напрямків $\bar{S}_1^0, \bar{S}_2^0, \dots, \bar{S}_n^0$, в кожному з яких послідовно знаходяться мінімальне значення, після чого система напрямків повертається так, щоб одна з осей вказували напрямок повного переміщених, а решта були ортогональні між собою.

Нехай $\bar{x}^0$ - 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 $f(\bar{x})$ в напрямках, відповідних $\bar{S}_1^0, \bar{S}_2^0, \dots, \bar{S}_n^0$, знаходячи послідовне наближення:

$\bar{x}_1^0 = \bar{x}_0^0 + \lambda_1 \bar{S}_1^0$, де $\lambda_1 = \ arg\min_{\lambda} f(\overline{x_0^0} + \lambda \bar{S}_1^0)$

…

$\bar{x}_n^0 = \bar{x}_{n-1}^0 + \lambda_n \bar{S}_n^0$, де $\lambda_n = \ arg\min_{\lambda} f(\overline{x_{n-1}^0} + \lambda \bar{S}_n^0)$

Наступна ітерація почнеться с точки $\bar{x}^1 = \ \bar{x}_n^0$

Якщо не змінити систему направлення, то ми будемо мати алгоритм циклічного по координатного спуску (Гаусу). Тому після завершення чергового k-го етапу обчислюємо нові напрямки пошуку. Ортогональні напрямки пошуку повертають так, щоб вони виявилися витягнутими уздовж "яру" і, таким чином, буде виключатися взаємодії змінних ($x_i x_j$). Напрямки пошуку витягуються уздовж головних осей квадратичної апроксимації цільової функції.

Розглянемо k-у ітерацію Розенброка. В результаті мінімізації по кожному 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 $\bar{A}_1^k, \bar{A}_2^k, \dots, \bar{A}_n^k$, що обчислюються за формулами такого вигляду:

$$\bar{A}_1^k = \lambda_1^k \bar{S}_1^k + \lambda_2^k \bar{S}_2^k + \cdots + \lambda_n^k \bar{S}_n^k;$$

$$\bar{A}_2^k = \lambda_2^k \bar{S}_2^k + \cdots + \lambda_n^k \bar{S}_n^k;$$

$$\dots;$$

$$\bar{A}_n^k = \lambda_n^k \bar{S}_n^k.$$

За допомогою отриманої системи векторів $\bar{A}_1^k, \bar{A}_2^k, \dots, \bar{A}_n^k$ будуємо нову систему ортогональних напрямків $\bar{S}_1^{k+1}, \bar{S}_2^{k+1}, \dots, \bar{S}_n^{k+1}$. При чому перший вектор спрямовується так, щоб він збігся з напрямком загального переміщених на k-му кроці, а інші напрямки виходять за допомогою процедури ортогоналізації Грама-Шмідта:

$$\bar{S}_1^{k+1} = \frac{\bar{A}_1^k}{\|\bar{A}_1^k\|};$$

$$\bar{B}_2^k = \bar{A}_2^k - \left[\left(\bar{A}_2^k\right)^T \bar{S}_1^{k+1}\right] \bar{S}_1^{k+1};$$

$$\bar{S}_2^{k+1} = \frac{\bar{B}_2^k}{\|\bar{B}_2^k\|}; \qquad (3.2.5.1)$$

$$\bar{B}_1^k = \bar{A}_1^k - \sum_{m=1}^{l-1} \left[\left(\bar{A}_l^k\right)^T \bar{S}_m^{k+1}\right] \bar{S}_m^{k+1};$$

$$\bar{S}_l^{k+1} = \frac{\bar{B}_l^k}{\|\bar{B}_l^k\|}; l = 2, \dots, n$$

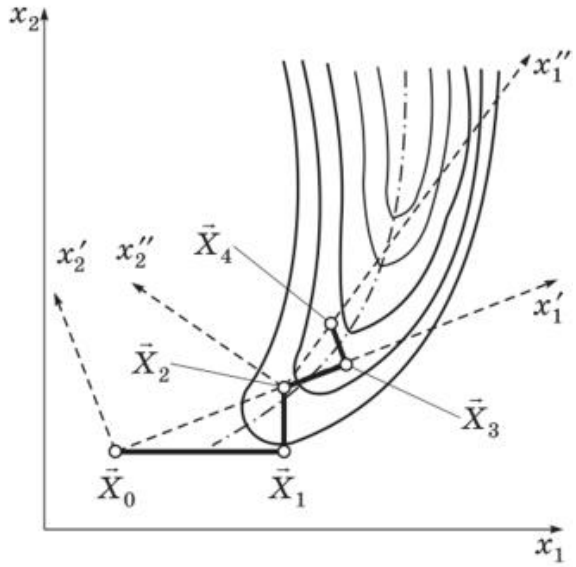


Рис. 3.2.5.1. Приклад використання методу Розенброка

Для ефективної роботи алгоритму необхідно, щоб жоден з векторів не став нульовим вектором $\bar{S}_1^{k+1}, \bar{S}_2^{k+1}, \dots, \bar{S}_n^{k+1}$. Для цього в алгоритмі слід розташовувати параметри $\lambda_1^k, \lambda_2^k, \dots, \lambda_n^k$ в порядку спадання за абсолютним значенням, $|\lambda_1^k| > |\lambda_2^k| > \cdots > |\lambda_n^k|$. Тоді якщо будь-які m з λ_i^k наближається до нуля, то відшукуються нові напрямки по співвідношенням (3.2.5.1) тільки для тих (n-m) напрямків, для яких $\lambda_i^k \neq 0$. Решта ж m напрямків залишаються незмінними: $\bar{S}_i^{k+1} = \bar{S}_i^k, i = \overline{(n-m+1), n}$. Так як перші (n-m) вектори взаємно ортогональні, $\lambda_i^k = 0, i = \overline{(n-m+1), n}$, перші (n-m) вектори не матимуть складові в напрямках $\bar{S}_i^{k+1}, i = \overline{(n-m+1), n}$. А оскільки ці останні напрямки взаємно ортогональні, то з цього слідує, що всі напрямки взаємно ортогональні.

Палмером було показано, що $\bar{B}_{j+1}^k$ і $\| \bar{B}_{j+1}^k \|$ пропорційні λ_j^k (при умові, що $\sum_{i=j}^n (\lambda_i^k)^2 \neq 0$). Отже, при обчисленні $\bar{S}_j^{k+1} = \bar{B}_j^k \Big/ \| \bar{B}_j^k \|$, величина $\lambda_j^k = 0$. Маючи на увазі, Палмер запропонував для обчислення $\bar{S}_j^{k+1}$ наступні співвідношення:

$$\bar{A}_i^k = \sum_{j=i}^{n} \lambda_j^k \cdot \bar{S}_j^{k+1}, i = 1, \dots, n,$$

$$\bar{S}_i^{k+1} = \frac{\bar{A}_i^k \cdot \|\bar{A}_i^k\|^2 - \bar{A}_{i-1}^{-k} \cdot \|\bar{A}_i^{-k}\|^2}{\|\bar{A}_{i-1}^{-k}\| \cdot \|\bar{A}_i^{-k}\| \cdot \left[\|\bar{A}_{i-1}^{-k}\|^2 - \|\bar{A}_i^{-k}\|^2\right]^{1/2}}, i = 2, \dots, n,$$

$$\bar{S}_1^{k+1} = \frac{\bar{A}_1^k}{\| \bar{A}_1^k \|}$$

Критерії зупинки алгоритму можуть бути стандартні (описані в попередніх алгоритмах прямих методів).

Для побудови системи ортогональних векторів для методу Розенброка може бути використано процес Грама-Шмідта, який полягає в проектуванні першого вектору базису на наступні за a_1 вектору a_i і знаходження ортогональних до цих проекцій векторів b_i.

Вважають $b_1 = a_1$, і якщо вже побудовані вектори $b_1, b_2, \dots, b_n$, то

$$b_i = a_i - \sum_{j=1}^{i-1} \frac{\langle a_i, b_j \rangle}{\langle b_i, b_j \rangle} b_j$$

Геометричний сенс описаного процесу полягає в тому, що на кожному послідовному кроці вектор b_i є перпендикуляром, встановленим до лінійної оболонки векторів $a_1, a_2, \dots, a_{i-1}$ до кінця вектора a_i.

Нормуються отримані вектори b_i,

$c_i = b_i / |b_i|$

отримують шукану ортонормовану систему $\{c_i\}$

У будь-якому евклідовому просторі завжди можна вибрати ортонормований базис $e_1, e_2, \dots, e_n$ при розкладанні векторів за яким:

$a = a_1 e_1 + a_2 e_2 + \cdots + a_n e_n$

$b = b_1 e_1 + b_2 e_2 + \cdots + b_n e_n$

Скалярний добуток буде виражено формулою

$\langle a, b \rangle = a^T b = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n.$

Алгоритм метода Розенброка з мінімізацією по напрямку

Початковий етап.

Нехай $\varepsilon > 0$ – скаляр, що використовується в критерію зупинки.

Обрати в якості $d_1 \dots dn$ координатні напрямки, початкову точку x_1, покласти

$y_1 = x_1$, $k = j = 1$ та перейти до основного етапу.

Основний етап. Крок 1.

Треба знайти *lymj* – оптимальне рішення задачі мінімізації

$f(y_j + lymj * d_j)$

при умові *lym* належить E^1 та покласти

$y_{j+1} = y_j + lymj * d_j$.

Якщо $j < n$, то замінити j на $j+1$ та повернутися до кроку 1.

В іншому випадку перейти до кроку 2.

Крок 2. Покласти $x_{k+1} = y_{n+1}$. Якщо $|x_{k+1} - x_k| < \varepsilon$, то зупинитися.

В іншому випадку покласти $y_1 = x_{k+1}$. Замінити k на $k+1$ покласти $j=1$ і перейти до кроку 3.

Крок 3. Побудувати нову множину лінійно-незалежних та взаємно ортогональних напрямків у відповідності з процедурою Грама-Шмідта. Визначити нові напрямки $d_1 \ldots dn$ та повернутися до кроку 1.

3.2.6. Метод спряжених напрямків

Вище було встановлено, що метод спуску по координатах навіть для квадратичної функції вимагає виконання нескінченної кількості ітерацій. Але якщо функція $F(\bar{r})$ є квадратичною функцією виду

$$F(\bar{r}) = (\bar{r}, A\bar{r}) + (\bar{b}, \bar{r}) + c \qquad (3.2.6.1)$$

з симетричною додатньо визначеною матрицею А, то можна побудувати такі напрямки спуску, що процес пошуку мінімуму буде реалізовано за скінченну кількість кроків [12; 15]. Особливо наглядно це для тривимірного простору, тобто для квадратичної функції двох аргументів $z = F(x, y)$. Геометричним образом квадратичної функції є поверхня другого порядку. Можна привести цю поверхню до канонічного вигляду, вибравши нову систему координат, центр якої співпадає з центром поверхні $F(\bar{r})$, а осі співпадають з головними висями еліпсоїдів рівня квадратичної форми. Очевидно, що достатньо виконати два спуски по координатах, щоб попасти в точку мінімуму (рис. 3.2.6.1).

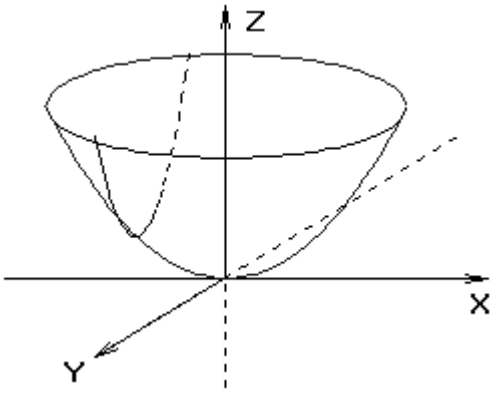


Рис. 3.2.6.1. Квадратична функція

Покажемо, що аналогічне твердження справедливе для додатньо визначеної квадратичної форми в N – вимірному евклідовому просторі R^N.

Означення. Дійсна симетрична матриця A порядку N називається додатньо визначеною, якщо $(\bar{x}, A\bar{x}) > 0$ для всіх ненульових векторів $\bar{x} \in R^N$.

Якщо є додатньо визначеною матриця A, то в просторі R^N можна ввести норму вектору по правилу

$$\|\bar{x}\|^2 = (\bar{x}, A\bar{x}) > 0 \text{ при } \bar{x} \neq 0. \qquad (3.2.6.2)$$

При цьому всі аксіоми для норми будуть виконані. Правило для норми (3.2.6.2) означає, що під скалярним добутком двох векторів $\bar{x}$ і $\bar{y}$ тепер слід розуміти величину $(\bar{x}, A\bar{y})$.

Означення. Якщо для векторів $\bar{x}$ і $\bar{y}$ виконується умова $(\bar{x}, A\bar{y}) = 0$, то такі вектори називаються спряженими по відношенню до даної матриці A.

Припустимо, що є деяка система попарно спряжених векторів

$$\bar{x}^{(1)}, \bar{x}^{(2)}, ..., \bar{x}^{(N)} \qquad (3.2.6.3)$$

Пронормуємо кожен із цих векторів у розумінні норми (3.2.6.2), тоді співвідношення між ними матимуть вигляд

$$\left(\bar{x}_i, A\bar{x}_j\right) = \delta_{ij}. \qquad (3.2.6.4)$$

Доведемо, що система взаємно спряжених векторів лінійно незалежна. Якщо припустити, що система векторів (3.2.6.3) лінійно залежна, то у її складі знайдеться вектор, який є лінійною комбінацією решти векторів. Тобто може

бути, що $\bar{x}_1 = \sum_{i=2}^{N} a_i \bar{x}_i$. Звідси одержуємо, що $(\bar{x}_1, A\bar{x}_1) = \sum_{i=2}^{N} a_i (\bar{x}_1, A\bar{x}_i) = 0$, але згідно з (3.2.6.4) $(\bar{x}_1, A\bar{x}_1) = 1$.

Одержане протиріччя доводить наше твердження і, отже, система N взаємно спряжених векторів є лінійно незалежною і може бути базисом в лінійному просторі R^N. Для даної матриці A існує нескінченна кількість базисів, які утворені із взаємно спряжених векторів.

Припустимо, що є деякий спряжений базис $\bar{x}_i$, $1 \le i \le N$. Виберемо довільну точку $\bar{r}$. Будь який рух із цієї точки можна розкласти по спряженому базису, тобто подати у вигляді

$$\bar{r} = \bar{r}_0 + \sum_{i=1}^{N} a_i x_1 \qquad (3.2.6.5)$$

Якщо вираз (3.2.6.5) підставити в (3.2.6.1) і виконати перетворення з урахуванням спряженості базису, то одержимо

$$F(\bar{r}) = F(\bar{r}_0) + \sum_{i=1}^{N} \left[a_i^2 + 2a_i (\bar{x}_i, A\bar{r}_0) + a_i (\bar{x}_i, \bar{b}) \right] \qquad (3.2.6.6)$$

Вираз (3.2.6.6) складається з доданків, кожен з яких відповідає лише одному доданку суми (3.2.6.5). Це означає, що рух по одному із спряжених напрямків $\bar{x}_1$ змінює лише один доданок у сумі (3.2.6.6), не змінюючи решти доданків. Будемо із $\bar{r}_0$ виконувати спуски до мінімуму по кожному із спряжених напрямків $\bar{x}_1$. Кожен спуск мінімізує свій доданок у сумі (3.2.6.6) і тому мінімум квадратичної функції буде точно досягнуто після виконання одного циклу спусків, тобто за N кроків.

Геометричний зміст спряженого базису полягає в тому, що якщо за вісі координат прийняти головні вісі еліпсоїдів рівня квадратичної функції, то один цикл спусків по цих координатах приводить точно в мінімум. Якщо перейти до деякої афінної системи координат, то функція залишиться квадратичною, зміняться лише її коефіцієнти. Положення головних осей в

вихідних афінних координатах буде деякою системою спряжених напрямків. При цьому в різних системах афінних координат будуть різні спряжені базиси.

Спряжений базис можна побудувати способом паралельних дотичних площин [14]. Нехай деяка пряма паралельна вектору $\bar{x}$, а квадратична функція (3.2.6.1) набуває на цій прямій мінімального значення в точці $\bar{r}_0$. Підставимо рівняння цієї прямої $\bar{r} = \bar{r}_0 + t\bar{x}$ в вираз (3.2.6.1) і будемо вимагати виконання умов мінімуму для функції

$$\varphi(t) = F(\bar{r}_0 + t\bar{x})$$

в точці $\bar{r} = \bar{r}_0$, тобто при $t = 0$.

Для цього скористаємося виразом (3.2.6.6), записавши в сумі лише один доданок

$$\varphi(t) = F(\bar{r}_0) + t^2 + t(\bar{x}, A\bar{r}_0 + \bar{b})$$

і покладемо $d\varphi/dt\,|_{t=0} = 0$.

Одержимо рівняння, якому задовольняє точка мінімуму:

$$(\bar{x}, 2A\bar{r}_0 + \bar{b}) = 0. \qquad (3.2.6.7)$$

Нехай на іншій прямій, паралельній вектору $\bar{x}$, функція (3.2.6.1) набуває мінімального значення в точці $\bar{r}_1$. Тоді аналогічним чином одержимо:

$$(\bar{x}, 2A\bar{r}_1 + \bar{b}) = 0. \qquad (3.2.6.8)$$

Віднімаючи від (3.2.6.8) вираз (3.2.6.7), одержимо

$$(\bar{x}, A(\bar{r}_1 - \bar{r}_0)) = 0.$$

Таким чином, встановлено, що напрямок, який з'єднує точки мінімуму на двох паралельних прямих є спряженим до напрямку цих прямих.

Отже, завжди можна побудувати вектор, спряжений довільному заданому вектору $\bar{x}$. Для цього в просторі R^N треба провести дві прямі, паралельні $\bar{x}$, і знайти на кожній прямій мінімум квадратичної функції (3.2.6.1). Вектор $\bar{r}_1 - \bar{r}_0$, який проходить через ці мінімуми, буде спряжений до

$\bar{x}$.

Припустимо, що є дві паралельні m-вимірні площини, утворені системою спряжених векторів $\bar{x}_i$, $1 \le i \le m \le N$. Нехай квадратична функція (3.2.6.1) набуває свого найменшого значення на цих площинах відповідно в точках $\bar{r}_0$ і $\bar{r}_1$.

Аналогічними міркуваннями можна довести, що вектор $\bar{r}_1 - \bar{r}_0$, який з'єднує точки мінімуму, буде спряженим до всіх векторів $\bar{x}_i$. Отже, якщо задана неповна система спряжених векторів $\bar{x}_i$, то цим способом завжди можна побудувати вектор $\bar{r}_1 - \bar{r}_0$, спряжений до всіх векторів цієї системи.

Розглянемо більш докладно один цикл процесу побудови спряженого базису. Нехай уже побудовано базис, в якому останні m векторів взаємно спряжені, а перші $n-m$ векторів не спряжені до останніх. Знайдемо мінімум квадратичної функції (3.2.6.1) в якій-небудь m – вимірній площині, утвореній останніми векторами базису. Через те, що ці вектори взаємно спряжені, то для цього достатньо довільно вибрати точку $\bar{r}_1$ і послідовно виконати із цієї точки m спусків по кожному із спряжених напрямків. Точку мінімуму в цій площині позначимо $\bar{r}_1$.

Тепер із точки $\bar{r}_1$ послідовно виконаємо спуски по перших $n-m$ векторах базису. Цей спуск виведе траєкторію із першої площини і ми попадемо в деяку точку $\bar{r}_2$. Із точки $\bar{r}_2$ знову виконаємо m спусків по кожному із спряжених напрямків і одержимо деяку точку $\bar{r}_3$. Точка $\bar{r}_3$ буде точкою мінімуму в площині, паралельній до площини, в якій лежить точка $\bar{r}_1$. Тому напрямок $\bar{r}_3 - \bar{r}_1$ буде спряженим до останніх m векторів базису. Якщо тепер в системі n–m неспряжених векторів замінити один вектор одержаним напрямком $\bar{r}_3 - \bar{r}_1$, то новий базис міститиме уже $m+1$ спряжений вектор. Обчислення можна почати з довільного базису, поклавши для нього $m=1$. Розглянутий алгоритм за один цикл обчислень збільшує кількість спряжених

векторів на одиницю, отже спряжений базис буде побудований за $n-1$ цикл.

Незважаючи на те, що поняття спряженого базису визначено лише для квадратичної функції, описаний процес побудований так, що його можна формально застосовувати до будь-якої функції. Зрозуміло, що при цьому мінімум вздовж напрямку треба знаходити, не використовуючи формул пов'язаних з конкретним видом квадратичної функції.

В малому околі мінімуму приріст достатньо гладкої функції можна представити у вигляді додатньо визначеної квадратичної форми. Якби така апроксимація була точною, то метод спряжених напрямків зійшовся б за скінчену кількість кроків. Через те, що апроксимація наближена, кількість кроків ітераційного процесу буде нескінченною. Швидкість збіжності цього методу в околі точки мінімуму буде квадратичною і завдяки цьому метод спряжених напрямків дозволяє знаходити мінімум функції n змінних з високою точністю.

В практичних обчисленнях, навіть для квадратичної функції, процес пошуку мінімуму не закінчується за n циклів. Побудова спряженого базису означає ортогоналізацію в метриці, породженій матрицею A. Всі процеси ортогоналізації дуже чутливі до похибок заокруглення. Якщо кількість змінних велика, то процес доводиться продовжувати за n циклів, замінюючи при цьому деякі вектори в уже побудованому спряженому базисі. Доцільно замінити той напрямок, при спуску вздовж якого функція найменше змінилася. Через те, що для довільної функції неможливо ввести поняття спряженості, то напрямок найменшої зміни функції замінюють незалежно від його номера в базисі. Іноді така заміна є корисною і для квадратичної функції.

Описаний вище алгоритм методу містить два спуски по спряжених напрямках і один спуск по неспряжених. Більш вигідним є цикл, в якому відразу після побудови нового спряженого напрямку вздовж нього виконують спуск з точки $\bar{r}_3$ і приходять в деяку точку $\bar{r}_4$. Тоді спуск із $\bar{r}_2$ в $\bar{r}_4$ буде спуском в площині всіх нових спряжених напрямків, тобто його можна

вважати початком нового циклу спусків. Тому із точки $\bar{r}_4$ відразу можна спускатися по неспряжених напрямках. При цьому новоодержаний напрямок розміщують в базисі на останньому місці, замінюючи той напрямок, на якому функція найменше змінилася при спуску з точки $\bar{r}_1$ в точку $\bar{r}_4$. Найменш вигідним може виявитися і новий напрямок, тоді наступний цикл виконують із старим базисом.

На сьогодні вважають, що метод спряжених напрямків є одним з найефективніших методів спуску. Його успішно використовують і при виродженому мінімумі, і при наявності слабо нахилених ділянок рельєфу – “плато” або “полонин”, і при розв’язуваних ярах, і при великій кількості змінних.

3.2.7. Метод паралельних дотичних

Цей метод використовує властивість квадратичної функції [12; 14], що полягає в тому, що будь-яка пряма, яка проходить через точку мінімуму функції x*, перетинає під рівними кутами дотичні до поверхонь рівного рівня функції в точках перетину

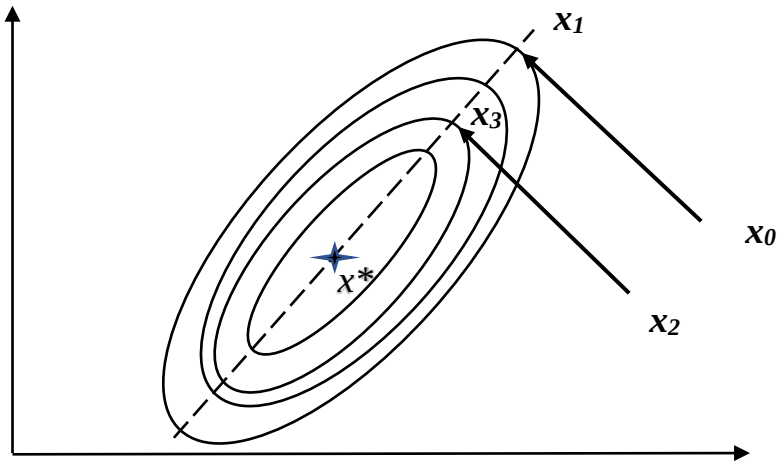


Рис. 3.2.7.1. Геометрична інтерпретація методу

Вибирається деяка початкова точка x_0 і виконується одновимірний пошук вздовж довільного напрямку, що приводить в точку x_1. Потім

обирається точка x_2, що не лежить на прямій за напрямком x_0 - x_1, і здійснюється одновимірний пошук вздовж прямої, паралельної x_0 - x_1. Отримана в результаті точка x_3 разом з точкою x_1 визначає напрямок x_1 - x_3 одновимірного пошуку, що дає точку мінімуму x^*.

У разі квадратичної функції n змінних оптимальне значення знаходиться за n ітерацій. Пошук мінімуму при цьому в кінцевому рахунку здійснюється у взаємно пов'язаних напрямках. У разі не квадратичної цільової функції напрямки пошуку виявляються пов'язаними щодо матриці Гессе.

Алгоритм методу паралельних дотичних полягає в наступному.

1. Оберемо початкову точку x_0.

За початкові напрямки пошуку приймають напрямки осей координат

$p_1, ..., p_0$,

тобто $p_i = e_i$ для $i = 1, ..., n$.

Тоді $e_i = (0, \dots, 0, 1, 0, \dots 0)^T$.

2. Виконують n одновимірних пошуків вздовж ортогональних напрямків p_i, $i = 1, ..., n$. При цьому кожен наступний пошук проводиться з точки мінімуму, отриманої на попередньому кроці. Величина кроку a_k знаходиться з умови

$f(x_k + a_k p_k) = \min f(x_k + a p_k)$.

Отриманий нову точку

$x_{k+1} = x_k + a_k p_k$.

3. Обирають новий напрямок $p = x_n - x_0$ та замінюють напрямки

$p_1, ..., p_n$ на $p_2, ..., pn$.

Таким чином $p_2, ..., p_n$ стають новими значеннями $p_1, ..., p_n$.

4. Здійснюють одновимірний пошук вздовж напрямку $p = p_n = x_n - x_0$. Змінюють x_0 на $x_{n+1} = x_n + a_n p_n$ та приймають цю точку за початкову точку x_0 для наступної ітерації. Далі переходимо до п. 1.

В результаті виконання розглянутої процедури здійснюється почергова заміна прийнятих спочатку напрямків пошуку. У підсумку після n кроків вони виявляться взаємно сполученими.

Питання до підрозділу 3.2.

1. Скільки вершин має гіперкуб?
2. Що таке регулярний симплекс?
3. Як будується регулярний симплекс?
4. Як будують “деформований” симплекс у методі Нелдера-Міда?
5. Коли спуск по координатах стає неможливим?
6. Коли спуск по координатах збігається лінійно?
7. Чим метод Хука-Дживса відрізняється від методу спуску по координатах?
8. Що є геометричним образом квадратичної функції?
9. Дайте геометричне тлумачення спряженим напрямкам для квадратичної функції 2-х аргументів?
10. Які вектори називають спряженими по відношенню до даної матриці?
11. Доведіть, що система взаємно спряжених векторів лінійно незалежна.
12. В чому полягає геометричний зміст спряженого базису?
13. Як можна побудувати вектор спряжений заданому вектору?

3.3. Градієнтні методи

3.3.1. Метод найшвидшого спуску

Вище було встановлено, що у методі покоординатного спуску процес пошуку екстремуму, наприклад мінімуму, зводиться до знаходження мінімуму функції однієї змінної у напрямку, паралельному одній з координатних висей. Але здійснювати спуск можна не лише паралельно висям координат. Якщо функція $F(\bar{r}) = F(x_1, x_2, ..., x_n)$ визначена в n - вимірному просторі, то вздовж будь-якої прямої лінії $\bar{r} = \bar{r}_0 + \bar{a}t$, заданої в

цьому просторі, функція $F(\bar{r}) = F(\bar{r}_0 + \bar{a}t)$ буде залежати лише від однієї змінної і мінімум функції на цій прямій можна знайти відомими методами.

Якщо відшукується мінімум функції $F(\bar{r})$ виходячи з деякої точки $\bar{r}_0$, то шукати цей мінімум доцільно у тому напрямку, на якому спостерігається найбільша зміна значень функції [13]. Такі міркування приводять до методу найшвидшого спуску, в якому вектор $\bar{a}$ вибирають по правилу

$$\bar{a} = -gradF(\bar{r})|_{\bar{r}=\bar{r}_0}$$

Тут $\bar{a}$ – вектор антиградієнта, тобто напрямок, у якому функція найбільше змінюється (зменшується) при нескінченно малому русі із даної точки. Спуск по цьому напрямку до мінімуму дає нове наближення – точку $\bar{r}_1$. В цій точці знову обчислюється градієнт і виконується наступний спуск.

Метод найшвидшого спуску набагато складніший за метод спуску по координатах через те, що для його реалізації треба обчислювати частинні похідні і градієнт [2; 10; 12]. Частинні похідні, як правило, доводиться обчислювати чисельно. По своїй швидкості цей метод не кращий за метод покоординатного спуску. Якщо траєкторія спуску попадає в істинний яр – спуск зупиняється, а при попаданні в розв'язуваний яр – траєкторія спуску стає звивистою, а сам спуск – дуже повільним.

Якщо функція $F(\bar{r})$ є додатньо визначеною квадратичною формою, тобто

$$F(\bar{r}) = (\bar{r}, A\bar{r}) + (\bar{b}, \bar{r}) + c, \qquad (3.3.1.1)$$

то процес найшвидшого спуску здійснюється по „точних" формулах [2; 3]. Дійсно, вздовж прямої лінії $\bar{r} = \bar{r}_n + \bar{a}t$ функція (3.3.1.1) квадратично залежить від параметра *t*:

$$\varphi(t) = F(\bar{r}_n + \bar{a}t) = F(\bar{r}_n) + (2A\bar{r}_n + \bar{b}, a)t + (\bar{a}, A\bar{a})t^2$$

З рівняння $\frac{d\varphi}{dt} = 0$ знаходимо її мінімум:

$$t = -\frac{(2A\bar{r}_n + \bar{b}, \bar{a})}{2(\bar{a}, A\bar{a})} \qquad (3.3.1.2)$$

Маючи точне значення *t*, можна знайти наступну точку спуску:

$$\bar{r}_{n+1} = \bar{r}_n + \bar{a}t, \quad F(\bar{r}_{n+1}) = F(\bar{r}_n) - \frac{(2A\bar{r}_n + \bar{b}, \bar{a})^2}{4(\bar{a}, A\bar{a})} \quad . \qquad (3.3.1.3)$$

Напрямок найшвидшого спуску визначається градієнтом квадратичної функції (3.3.1.1):

$$\bar{a} = -gradF(\bar{r})|_{\bar{r}=\bar{r}_n} = -(2A\bar{r}_n + \bar{b})$$

Підставляючи вираз для вектору $\bar{a}$ у формули (3.3.1.2) і (3.3.1.3), одержимо “точні” вирази для побудови послідовності спусків.

Якщо квадратичну форму (3.3.1.1) записати у новій системі координат, осі якої направлені по власних векторах матриці *A*, і в цій системі координат проаналізувати весь процес спуску, то можна довести, що для квадратичної функції метод найшвидшого спуску сходиться лінійно і справедлива оцінка

$$|r_{n+1} - r| \le q|r_n - r|,$$

$$q = \frac{\lambda_{\max} - \lambda_{\min}}{\sqrt{\lambda_{\max}^2 + \lambda_{\min}^2}} < 1. \qquad (3.3.1.4)$$

Тут λ – власні значення додатньо визначеної матриці *A*. Якщо $\lambda_{\min} << \lambda_{\max}$, то лініями рівня квадратичної форми будуть сильно витягнуті еліпси. З формули (3.3.1.4) витікає, що у цьому випадку $q \approx 1$ і збіжність методу буде дуже повільною.

Якщо лініями рівня квадратичної форми є дуже витягнуті еліпси, то фактично околом точки екстремуму буде розв’язуваний яр (або гребінь). При невдалому виборі вихідної точки $\bar{r}_0$ (рис. 3.3.1.1), може виникнути ситуація найгіршої збіжності методу, коли рух до мінімуму буде проходити по дну дуже витягнутого яру.

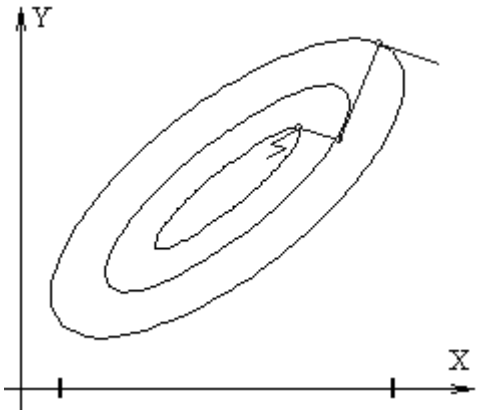


Рис. 3.3.1.1. Метод найшвидшого спуску

Можлива наступна модифікація методу найшвидшого спуску [13; 15]. Будемо виконувати по напрямку, протилежному градієнту, лише нескінченно малий крок і після нього знову уточнювати напрямок спуску. Тоді будемо рухатися по лінії $\bar{r}(t)$, яка буде розв'язанням системи звичайних диференціальних рівнянь

$$\frac{dr}{dt} = -gradF(\bar{r}(t)). \qquad (3.3.1.5)$$

Вздовж цієї лінії $dF/dt = (dF/dr)(dr/dt) = -(gradF)^2 < 0$ тобто, функція зменшується і відбувається рух до мінімуму при $t \to \infty$.

У тривимірному просторі система диференціальних рівнянь (3.3.1.5) описує траєкторію руху матеріальної точки, яка „скочується" по поверхні $F(\bar{r})$ до точки мінімуму. Реалізація такого варіанту методу найшвидшого спуску вимагає використання методів чисельного інтегрування систем звичайних диференціальних рівнянь. Зрозуміло, що якщо рельєф функції має звивисті яри, то збіжність методу буде поганою.

На сьогодні алгоритми методу найшвидшого спуску і всіх його модифікацій ще недостатньо відпрацьовані. Тому цей метод рідко використовують при розв'язуванні складних нелінійних задач з великою кількістю змінних.

3.3.2. Метод ярів

Припустимо, що розв'язується задача $F(\bar{r}) = \min$ і по ходу процесу розв'язування видно, що ми попали в яр. Для того, щоб швидко пройти яр, треба визначити його напрямок. Виберемо довільну точку $\bar{p}_0$ і виконаємо із неї спуск на дно яру, не вимагаючи високої точності. Кінцеву точку спуску позначимо $\bar{q}_0$. Тепер виберемо точку $\bar{p}_1$, розташовану недалеко від $\bar{p}_0$ і виконаємо спуск із цієї точки. Попадемо в деяку точку $\bar{q}_1$ на дні яру. Проведемо через точки $\bar{q}_0$ і $\bar{q}_1$ пряму лінію, яка вказує приблизний напрямок дна яру. Змістимося по цій лінії в сторону зменшення функції і виберемо нову точку $\bar{p}_2$ по правилу

$$\bar{p}_2 = \bar{q}_1 \pm (\bar{q}_1 - \bar{q}_0)h.$$

У цій формулі вибирається „+", якщо $F(\bar{q}_1) < F(\bar{q}_0)$ і „–" в протилежному випадку, так що рухатися будемо в сторону зниження дна яру. Крок руху по яру (величина h) визначається експериментально.

Через те, що дно яру звивисте, точка $\bar{p}_2$ лежатиме не на дні яру, а на його схилі. Із цієї точки знову виконаємо спуск на дно яру і попадемо в деяку точку $\bar{q}_2$. З'єднавши точки $\bar{q}_1$ і $\bar{q}_2$ прямою лінією, визначимо новий напрямок дна яру і виконаємо новий крок по яру (рис. 3.3.2.1).

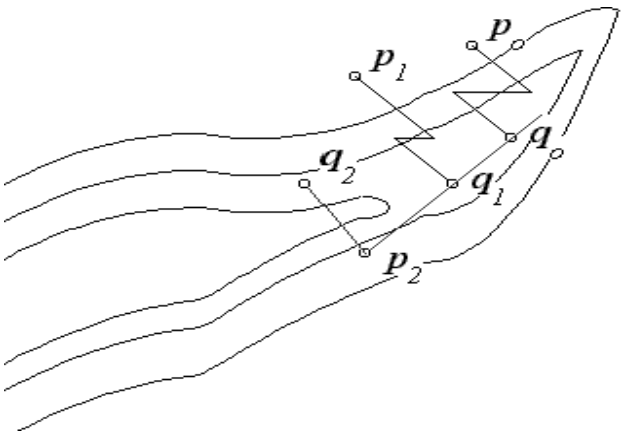


Рис. 3.3.2.1. Метод ярів

Процес слід продовжувати доти, доки значення функції на дні яру, тобто

в точках $\bar{q}_n$, зменшуються. Процес закінчується, коли

$F(\bar{q}_{n+1}) > F(\bar{q}_n)$.

Таким чином, метод ярів дозволяє пройти вздовж яру і вийти в котловину навколо точки мінімуму. В цій котловині значення мінімуму можна уточнити іншими методами. Метод ярів дозволяє знаходити мінімуми досить складних функцій від 5-10 змінних. Але цей метод не є універсальним і вимагає відстежування процесу пошуку мінімуму і внесення необхідних коректив в алгоритм з боку програміста.

У наведеному вище викладі метод ярів – чисто емпіричний метод. Якщо виконати математичне обґрунтування цього методу, то одержимо метод Левенберга-Марквардта.

3.3.3. Метод Флетчера-Рівса

Для квадратичної функції N змінних з симетричною додатньо визначеною матрицею A послідовні N спусків вздовж спряжених напрямків приводять в точку мінімуму.

Якщо при побудові спряжених напрямків для квадратичної цільової функції вектори вихідної системи вибирати певним чином, то можна одержати ряд переваг [14; 15]. Нехай квадратична функція задана у вигляді

$$F(\bar{r}) = \frac{1}{2}(\bar{r}, A\bar{r}) + (\bar{b}, \bar{r}) + c.$$

Тоді градієнт $F(\bar{r})$ обчислюється по формулі $\nabla F(\bar{r}) = A\bar{r} + \bar{b}$ і справедливі формули

$$\nabla F(\bar{r}^{(1)}) = A\bar{r}^{(1)} + \bar{b},$$

$$\nabla F(\bar{r}^{(0)}) = A\bar{r}^{(0)} + \bar{b},$$

$$\nabla F(\bar{r}^{(1)}) - \nabla F(\bar{r}^{(0)} = A(\bar{r}^{(1)} - \bar{r}^{(0)}).$$

Систему спряжених напрямків у методі Флетчера–Рівса будують так:

- фіксуємо деяку точку $\bar{r}^{(0)}$,

- покладемо $\bar{x}^{(0)} = -\nabla F(\bar{r}^{(0)})$.

Виконаємо спуск із точки $\bar{r}^{(0)}$ у напрямку $\bar{x}^{(0)}$ і знайдемо точку $\bar{r}^{(1)} = \bar{r}^{(0)} + t_1 \bar{x}^{(0)}$ в якій функція $F(\bar{r})$ має мінімум.

- визначимо напрямок $\bar{x}^{(1)} = -\nabla F(\bar{r}^{(1)}) + \omega_1 \bar{x}^{(0)}$ так, щоб $\bar{x}^{(1)}$ і $\bar{x}^{(0)}$ були спряженими, тобто виконувалася умова $(\bar{x}^{(0)}, A\bar{x}^{(1)}) = 0$. Для цього виконаємо наступні дії

$$\nabla F(\bar{r}^{(1)}) - \nabla F(\bar{r}^{(0)}) = A(\bar{r}^{(1)} - \bar{r}^{(0)}) = t_1 A\bar{x}^{(0)}.$$

Транспонуємо цей вираз і домножимо його на A^{-1} справа, будемо мати:

$$(\nabla F(\bar{r}^{(1)}) - \nabla F(\bar{r}^{(0)}))^T A^{-1} = t_1 \bar{x}^{(0)^T} A^T A^{-1}$$

$$\bar{x}^{(0)^T} = \frac{(\nabla F(\bar{r}^{(1)}) - \nabla F(\bar{r}^{(0)}))^T A^{-1}}{t_1}$$

Підставимо одержаний вираз в умову спряженості $(\bar{x}^{(0)}, A\bar{x}^{(1)}) = 0$

$$\frac{(\nabla F(\bar{r}^{(1)}) - \nabla F(\bar{r}^{(0)}))^T A^{-1} A\bar{x}^{(1)}}{t_1} = 0$$

Якщо замість $\bar{x}^{(1)}$ підставити його вираз, то одержимо

$$(\nabla F(\bar{r}^{(1)}) - \nabla F(\bar{r}^{(0)}))^T (-\nabla F(\bar{r}^{(1)} - \omega_1 \nabla F(\bar{r}^{(0)})) = 0$$

Через те, що по побудові, градієнти в точках $\bar{r}^{(0)}$ і $\bar{r}^{(1)}$ ортогональні, тобто скалярний добуток $(\nabla F(\bar{r}^{(0)}, \nabla F(\bar{r}^{(1)})) = 0$, з останнього виразу одержимо формулу Флетчера-Рівса для ω_1 у вигляді

$$\omega_1 = \left\| \nabla F(\bar{r}^{(1)} \right\|^2 / \left\| \nabla F(\bar{r}^{(0)} \right\|^2.$$

На k – ій ітерації ми уже матимемо спряжені напрямки $\bar{x}^{(0)}, \bar{x}^{(1)} \ldots \bar{x}^{(k-1)}$ і напрямок $\bar{x}^{(k)}$ буде обчислюватися по формулі

$$\bar{x}^{(k)} = -\nabla F(\bar{r}^{(0)}) + \omega_1 \bar{x}^{(0)} + \omega_2 \bar{x}^{(1)} + \ldots + \omega_k \bar{x}^{(k-1)} \quad \text{і}$$

$$\omega_k = \frac{\left\| \nabla F(\bar{r}^{k}) \right\|^2}{\left\| \nabla F(\bar{r}^{(k-1)}) \right\|^2}.$$

Вираз для $\bar{x}^{(k)}$ можна переписати у більш зручному ітеративному вигляді

$$\bar{x}^{(k)} = -\nabla F(\bar{r}^{(k)}) + \omega_k \bar{x}^{(k-1)}.$$

Таким чином, в методі Флетчера-Рівса при побудові системи спряжених напрямків за рахунок спеціального вибору вихідної системи лінійно незалежних векторів ніде явно не використовується матриця квадратичної форми і тому метод легко узагальнюється для випадку мінімізації будь яких функцій. Для неквадратичних функцій N змінних метод стає ітераційним. Флетчер і Рівс рекомендують циклічно повторювати алгоритм методу після виконання кожних $N + 1$ кроків [13].

Алгоритм методу Флетчера-Рівса.

Крок 1.

- задати вихідну точку $\bar{r}^{(0)}$;
- обчислити $\nabla F(\bar{r}^{(0)})$;
- покласти $\bar{x}^{(0)} = -\nabla F(\bar{r}^{(0)})$;
- покласти $k = 0$;

Крок 2.

З точки $\bar{r}^{(k)}$ виконати спуск у напрямку $\bar{x}^{(k)}$ і знайти точку $\bar{r}^{(k+1)}$ в якій цільова функція набуває мінімального значення. Спуск виконується одним із відомих методів одновимірної оптимізації.

Крок 3.

- обчислити $\nabla F(\bar{r}^{(k+1)})$;

– обчислити $\left\|\nabla F(\bar{r}^{(k)})\right\|, \left\|\nabla F(\bar{r}^{(k+1)})\right\|$ і ω_{k+1};

– якщо норма градієнта $\left\|\nabla F(\bar{r}^{(k+1)})\right\| \le Eps$, завершити процес.

– побудувати новий спряжений напрямок $\bar{x}^{(k+1)}$;

– покласти $k = k+1$, якщо $k \le N$, де N – кількість змінних задачі, то перейти на Крок 2, інакше покласти $k = 0$, прийняти $\bar{r}^{(N)}$, за $\bar{r}^{(0)}$ і перейти на Крок 1.

3.3.4. Метод Девідона-Флетчера-Пауелла

Метод Девідона-Флетчера-Пауелла побудовано таким чином, що при його реалізації не треба обчислювати обернену матрицю Гессе [2; 3; 13; 15]. Матриця напрямків A обчислюється так, щоб для квадратичної цільової функції після виконання n кроків вона дорівнювала H^{-1}. На початку процесу вихідну матрицю вибирають у вигляді одиничної матриці $A^{(0)} = I$, тому вихідним напрямком мінімізації буде напрямок найшвидшого спуску. Але це не обов'язково, вихідною матрицею може бути будь-яка симетрична додатньо визначена матриця. При реалізації алгоритму відбувається поступовий перехід від напрямку найшвидшого спуску до напрямку, який визначається методом Ньютона і на відповідних етапах мінімізації використовуються переваги кожного з цих методів.

Алгоритм методу Девідона-Флетчера-Пауелла.

На початку процесу задані вихідна точка $\bar{r}^{(0)}$ і парметри завершення $\varepsilon_1, \varepsilon_2$.

1. Задати початкову точку $\bar{r}^{(k)}$, $k = 0$. Обчислити $\nabla f(\bar{r}^{(k)}$, покласти $A^{(0)} = I$.

2. Обчислити параметр $\lambda^{*(k)}$, виконавши одновимірний спуск виду $f[\bar{r}^{(k)} - \lambda^{*(k)} A(\bar{r}^{(k)}) \nabla f(\bar{r}^{(k)})]$.

3. Обчислити точку $\bar{r}^{(k+1)}$ по формулі

$\bar{r}^{(k+1)} = \bar{r}^{(k)} - \lambda^{*(k)} A(\bar{r}^{(k)}) \nabla f(\bar{r}^{(k)})$.

4. Обчислити $f(\bar{r}^{(k+1)})$, $\nabla f(\bar{r}^{(k+1)})$,

$\Delta g^{(k)} = \nabla f(\bar{r}^{(k+)}) - \nabla f(\bar{r}^{(k)})$, $\Delta \bar{r}^{(k)} = \bar{r}^{(k+1)} - \bar{r}^{(k)}$.

5. Перевірити виконання двох наступних нерівностей

$$\left| \frac{f(\bar{r}^{(k+1)}) - f(\bar{r}^{(k)})}{f(\bar{r}^{(k)})} \right| > \varepsilon_1, \text{ або } \left| \Delta g^{(k)} > \varepsilon_1 \right|, \text{ якщо } f(\bar{r}) \to 0$$

$$\left| \frac{\Delta r_i^{(k)}}{r_i^{(k)}} \right| > \varepsilon_2, \text{ або } \Delta r_i^{(k)} > \varepsilon_2, \text{ якщо } r_i \to 0.$$

6. Якщо нерівності не виконуються, то процес закінчено, інакше викононується пункт 7.

7. $k = k + 1$. Обчислити матрицю $A^{(k)}$:

$$A^{(k)} = A^{(k-1)} + \frac{\Delta \bar{r}^{(k-1)} \Delta \bar{r}^{(k-1)^T}}{\Delta \bar{r}^{(k-1)^T} \Delta \bar{g}^{(k-1)}} - \frac{A^{(k-1)} \Delta \bar{g}^{(k-1)} \Delta \bar{g}^{(k-1)^T} A^{(k-1)}}{\Delta \bar{g}^{(k-1)^T} A^{(k-1)} \Delta \bar{g}^{(k-1)}}.$$

Перейти на пункт 2.

Остання формула зберігає властивості симетрії і додатньої визначеності матриць і, таким чином, алгоритм забезпечує зменшення цільової функції від ітерації до ітерації.

Питання до підрозділу 3.3.

1. Метод найшвидшого сруску.

2. У чому складність застосування методу найшвидшого спуску?

3. Метод ярів

4. В яких умовах з методу ярів утворюється метод Левенберга-Марквардта?

5. Метод Флетчера-Рівса.

6. Метод Девідона-Флетчера-Пауелла.

3.4. Методи другого порядку

3.4.1. Метод Ньютона

В методі Ньютона при побудові траєкторії спуску використовують значення перших і других частинних похідних цільової функції [2; 3; 13]. Для побудови ітераційного процесу виконаємо розклад цільової функції в ряд Тейлора

$$F(\bar{r})=F(\bar{r}^{(k)})+\nabla F(\bar{r}^{(k)})^T\,\Delta\bar{r}+\frac{1}{2}\Delta\bar{r}^T\nabla^2F(\bar{r}^{(k)})\Delta\bar{r}+\ldots.$$

і побудуємо квадратичну апроксимацію функції $F(\bar{r})$ в околі точки $\bar{r}^{(k)}$.

$$ApF(\bar{r};\bar{r}^{(k)})=F(\bar{r}^{(k)})+\nabla F(\bar{r}^{(k)})^T\,\Delta\bar{r}+\frac{1}{2}\Delta\bar{r}^T\nabla^2F(\bar{r}^{(k)})\Delta\bar{r};$$

Для переходу із точки $\bar{r}^{(k)}$ в точку $\bar{r}^{(k+1)}$ виберемо прирости аргументів $\Delta\bar{r}$ так, щоб градієнт апроксимуючої функції ставав нулем в точці $\bar{r}^{(k+1)}$, тобто

$$\nabla ApF(\bar{r};\bar{r}^{(k)})=\nabla F(\bar{r}^{(k)})+\nabla^2F(\bar{r}^{(k)})\Delta\bar{r}=0,$$

звідки одержуємо

$$\Delta\bar{r}=-[\nabla^2F(\bar{r}^{(k)})]^{-1}\nabla F(\bar{r}^{(k)})$$

і точка $\bar{r}^{(k+1)}$ визначається за формулою

$$\bar{r}^{(k+1)}=\bar{r}^{(k)}-[\nabla^2F(\bar{r}^{(k)})]^{-1}\nabla F(\bar{r}^{(k)}).$$

3.4.2. Метод Левенберга-Марквардта

Практика використання методів найшвидшого спуску і Ньютону показує, що обидва методи мають свої недоліки і переваги [14; 16]. Метод найшвидшого спуску погано працює в безпосередньому околі стаціонарної точки, коли компоненти градієнта цільової функції стають малими. Метод Ньютона ефективний саме в такій ситуації. Спроба поєднати переваги обох

методів виконана в методі Левенберга, в якому ітераційний процес побудови траєкторії спуску здійснюється по формулі

$$\bar{r}^{(k+1)} = \bar{r}^{(k)} - (H + \lambda I)^{-1} \nabla F(\bar{r}^{(k)})$$

в якій λ - число (параметр), H – матриця Гессе, I – одинична матриця.

Вибором величини параметра λ надається перевага тому, чи іншому методу. Якщо λ велике, то

$$(H + \lambda I)^{-1} \approx (\lambda I)^{-1} = \left(\frac{1}{\lambda}\right) I$$

і ітераційний процес здійснюється по методу найшвидшого спуску, якщо ж λ мале, то по методу Ньютона. При реалізації метода Левенберга треба стежити за величинами зміни значень цільової функції. На початку ітераційного процесу задається велике значення λ і спуск виконується у напрямку антиградієнта. Якщо ж, починаючи з деякого кроку ітераційного процесу цільова функція змінюється мало, то доцільно зменшити величину λ і продовжити процес по методу Ньютона.

Якщо цільова функція має складний рельєф і в процесі пошуку її мінімуму ми попадаємо в довгу і вузьку “западину”, то компоненти градієнта направлені вздовж “дна” западини стають малими, а компоненти градієнта, направлені до “стінок” западини, зростають. При використанні методу найшвидшого спуску траєкторія спуску стає звивистою, ламаною і процес спуску уповільнюється. Зрозуміло, що в такій ситуації треба виконувати більші кроки вздовж “дна” западини і менші у напрямку до її “стінок”, тобто компоненти градієнта доцільно “підправити” з урахуванням кривини поверхні цільової функції. Це реалізовано в методі Левенберга-Марквардта (його ще називають *алгоритм LMA*), в якому ітераційний процес побудови траєкторії спуску виконується по формулі

$$\bar{r}^{(k+1)} = \bar{r}^{(k)} - (H + \lambda\, diag\,[H])^{-1} \nabla F(\bar{r}^{(k)}).$$

В цій формулі одинична матриця замінюється на діагональ матриці Гессе. Через те, що матриця Гессе визначає кривину поверхні в околі точки $\bar{r}^{(k)}$, метод дозволяє одержувати більший крок у напрямку в якому компоненти градієнта малі, і менший крок у напрямку “стінок” западини.

Алгоритм LMA особливо успішно використовується при розв’язуванні задач нелінійної мінімізації методом найменших квадратів, тобто у випадку коли цільова функція має вигляд

$$F(\bar{p})=\frac{1}{2}\sum_{i=1}^{m} r_i^2(\bar{p})$$

де $\bar{p}=(p_1\, p_2 \ldots p_n)$ – вектор параметрів, які треба знайти, а $r_i(\bar{p})$ – розбіжності, зумовлені вектором $\bar{p}$ та значеннями $m \geq n$.

Питання до підрозділу 3.4.

1. На що вказує вектор градієнта (антиградієнта)?

2. Коли градієнтні методи працюють повільно?

3. Чим визначається швидкість збіжності градієнтного методу для додатньо визначеної квадратичної функції?

4. Як можна швидко пройти яр?

5. Чому метод Флетчера-Рівса називають ще методом спряжених градієнтів?

6. Як використовується матриця Гессе у методі Ньютона?

7. Які умови накладено на градієнт апроксимуючої функції при побудові методу Ньютона?

8. Коли метод найшвидшого спуску переходить в метод Ньютона в методі Девідона-Флетчера-Пауелла?

9. Які недоліки і які переваги мають метод найшвидшого спуску і метод Ньютона?

10. Які переваги методу найшвидшого спуску і методу Ньютона поєднуються у методі Левенберга-Марквардта?

3.5. Методи врахування обмежень

Найбільш розповсюдженими задачами на практиці є оптимізаційні задачі при наявності обмежень, тобто задачі пошуку оптимального результату, що задовольняє певну систему обмежень. Існування ефективних методів вирішення безумовних задач оптимізації завжди підштовхує на спробу використання цих методів для розв'язання задач з умовами, після відповідного перетворення задачі з умовою в еквівалентну за результатами безумовну задачу.

Нехай необхідно знайти рішення задачі мінімізації виду

$$f^*(x)=min\left\{f(\bar{x})\middle|h_j(\bar{x})=0, j=\overline{1,m} g_j(\bar{x})\le 0, j=\overline{m+1,k}\right\}, \qquad (3.5.1)$$

в якій цільова функція і функції системи обмежень являють собою випуклі, як правило, функції.

Ідея методу штрафних функцій полягає у наступному [14;15]. Будують таку допоміжну функцію

$$\varphi(\bar{x},\bar{r}) = f(\bar{x}) + \sum_{j=1}^{m} r_j \cdot H\left[h_j(\bar{x})\right] + \sum_{j=m+1}^{k} r_j \cdot G\left[g_j(\bar{x})\right]. \qquad (3.5.2)$$

Ця функція побудована таким чином, що рішення для $\varphi(\bar{x},\bar{r})$ за своїм значенням буде найближчим до припустимого, з точки зору обмежень, рішення для f(x).

Тобто знаходимо рішення задачі для f* як рішення задачі методами безумовної оптимізації

$$min\ \varphi(\bar{x},\bar{r}) \qquad (3.5.3).$$

3.5.1. Зовнішні штрафні функції

В методі зовнішніх штрафних функцій функції H(·) і G(·) вибирають таким чином, щоб вони ставали відмінними від нуля (додатними) при порушенні відповідного обмеження. А так як ми мінімізуємо (3.5.2) то рух в

бік обмеження стає невигідним. В даному методі функції H(·) і G(·) всередині допустимої області мають бути рівними нулю. Наприклад для обмежень нерівностей:

$G_j\left[g_j(\bar{x})\right] \to 0$ при умові, що

$g_j(\bar{x}) \to 0^*$

Наближений розв'язок задачі (3.5.1) отримується в результаті розв'язку послідовності задач (3.5.3) при $r_j \to \infty, j = \overline{1,k}$. Відповідні методи ще називають методами зовнішньої точки.

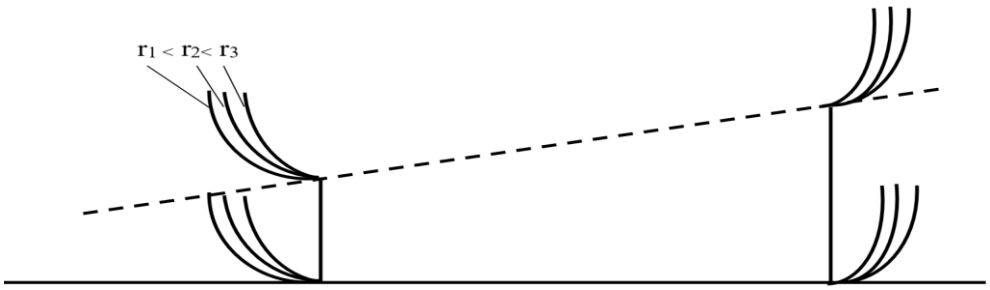


Рис. 3.5.1.1. Зовнішні штрафні функції

3.5.2. Бар'єрні функції

В методі бар'єрних функцій [14; 15] функції H(·) і G(·) в допустимій області вибираються відмінними від нуля, притому такими, щоб при наближенні до границі допустимої області (з середини) вони зростали, перешкоджаючи виходу при пошуку за границю області.

В цьому випадку ці функції повинні приймати малі (додатні чи від'ємні) значення всередині допустимої області і великі додатні поблизу границі (всередині області). Наприклад для обмежень нерівностей:

$G_j\left[g_j(\bar{x})\right] \to 0$ при $g_j(\bar{x}) \to 0^*$.

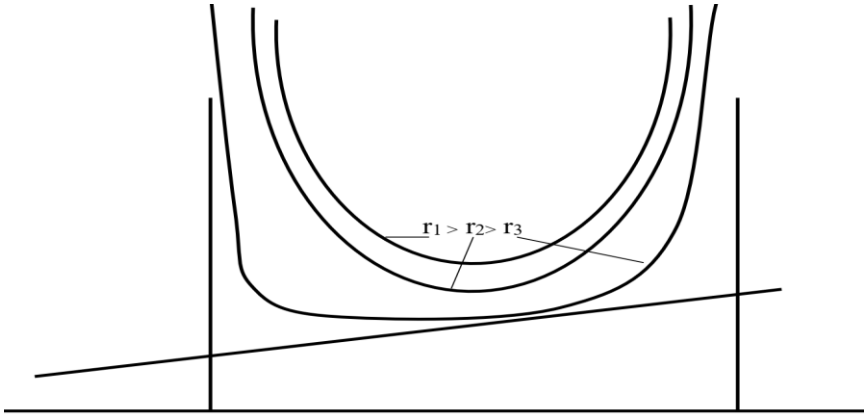


Рис. 3.5.2.1. Бар’єрні функції

Такого типу методи називають ще методами внутрішньої точки. В алгоритмах, що використовують функції штрафу даного типу бар’єрні функції, потребують, щоб в процесі пошуку точка $\bar{x}$ завжди залишалась внутрішньою точкою допустимої області.

Наближений результат задачі (3.5.1) отримується в результаті розв’язку послідовності задач виду (3.5.3) при $r_j \to 0,\ j = \overline{1,k}$.

При виборі функцій штрафів для обмеження тотожності зазвичай вимагають щоб

$$H_j\left[h_j(\bar{x})\right] \to 0 \quad \text{при} \quad h_j(\bar{x}) \to 0$$

Це можуть бути, наприклад, функції наступного виду:

$H_j\left[h_j(\bar{x})\right] = \left|h_j(\bar{x})\right|$.

$H_j\left[h_j(\bar{x})\right] = \left|h_j(\bar{x})\right|^2$.

$H_j\left[h_j(\bar{x})\right] = \left|h_j(\bar{x})\right|^{\alpha}$, при парному ступені α.

Для обмежень нерівностей функції штрафу підбирають таким чином, щоб

$G_j\left[g_j(\bar{x})\right]=0$ при $g_j(\bar{x})\le 0$;

$G_j\left[g_j(\bar{x})\right]>0$ при $g_j(\bar{x})>0$.

Цій вимогі відповідають, наприклад, функції виду:

1) $G_j\left[g_j(\bar{x})\right]=\frac{1}{2}\left\{g_j(\bar{x})+\left|g_j(\bar{x})\right|\right\}$,

2) $G_j\left[g_j(\bar{x})\right]=\left[\frac{1}{2}\left\{g_j(\bar{x})+\left|g_j(\bar{x})\right|\right\}\right]^2$,

3) $G_j\left[g_j(\bar{x})\right]=\left[\frac{1}{2}\left\{g_j(\bar{x})+\left|g_j(\bar{x})\right|\right\}\right]^\alpha$, при парному ступені α.

В якості бар'єрних функцій для обмежень нерівностей можуть служити, наприклад, функції виду:

$G_j\left[g_j(\bar{x})\right]=-\frac{1}{g_j(\bar{x})}$,

$G_j\left[g_j(\bar{x})\right]=-\ln\left[-g_j(\bar{x})\right]$.

Послідовність дій при реалізації методів штрафних чи бар'єрних функцій виглядає наступним чином:

На основі задачі (3.5.1) будуємо функцію (3.5.2).

Вибираємо початкове наближення $\bar{x}$ і початкові значення коефіцієнтів штрафу r_j.

Розв'язуємо задачу (3.5.3).

Якщо отриманий розв'язок не задовольняє системі обмежень, то у випадку використання методу штрафних функцій збільшуємо значення коефіцієнтів штрафу r_j і знову розв'язуємо задачу (3.5.3).

У випадку методу бар'єрних функцій, щоб можна було отримати розв'язок на границі, значення коефіцієнтів r_j зменшуються.

Процес зупиняється, якщо знайдений розв’язок задовольняє системі обмежень з заданою точністю.

Питання до підрозділу 3.5.

1. Яка ідея методу штрафних функцій?
2. В чому особливість методу бар’єрних функцій?
3. Які є види функцій штрафу?
4. Які є види бар’єрних функцій?
5. У чому різниця штрафних функцій та бар’єрних функцій?

РОЗДІЛ 4. МЕТОДИ ВИПАДКОВОГО ПОШУКУ

Статистичні методи або методи випадкового пошуку отримали достатньо широке поширення при побудові оптимальних рішень в різних додатках [17; 18]. Це пояснюється в тим, що із зростанням розмірності задач різко знижується ефективність регулярних методів пошуку (детермінованих): так зване «прокляття розмірності».

Часто інформація про оптимізованого об’єкта занадто мала для того, щоб можна було застосувати детерміновані методи. Досить часто статистичні алгоритми використовують при пошуку оптимального рішення в системах управління (рис. 4.1), коли відгук системи можна отримати тільки при завданні управляючих впливів $\bar{X}$ на її входах. В таких ситуаціях статистичні алгоритми можуть виявитися значно ефективніше детермінованих.

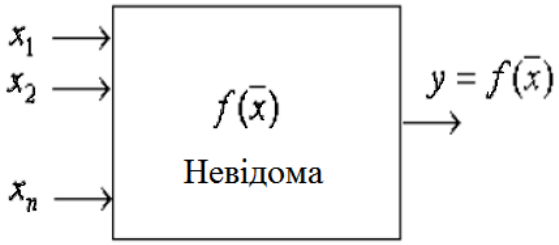


Рис. 4.1. Оптимальне управління системою

Найбільший результативним застосування статистичних методів є при вирішенні задач великої розмірності або при пошуку глобального екстремуму.

Під випадковими або статистичними методами пошуку будемо розуміти методи, які використовують елемент випадковості або при зборі інформації про цільову функцію при пробних кроках, або для покращення значень функції при робочому кроці. Випадковим чином може вибиратися напрямок спуску, довжина кроку, величина кроку при порушенні обмеженні і т.д.

Статистичні алгоритми мають ряд переваг:

- простотою реалізації і налагодженню програм;
- надійністю;
- універсальністю;
- можливістю введення операцій навчання в алгоритм пошуку;
- можливістю введення операцій прогнозування оптимальної точки (оптимального рішення).

А основними недоліками є велика кількість обчислень мінімізованої функції і повільна збіжність в районі екстремуму.

Прийнято вважати, що перевага статистичних методів проявляється зростанням розмірності задач, тому що обчислювальні витрати в детермінованих методах пошуку з ростом розмірності ростуть швидше, ніж в статистичних алгоритмах.

Розрізняють спрямований та неспрямований випадковий пошук [19].

Неспрямований випадковий пошук. При такому пошуку всі наступні випробування проводять абсолютно незалежно від результатів попередніх. Збіжність такого пошуку дуже мала, але є важлива перевага, пов'язана з можливістю вирішення задач (шукати глобальний екстремум). Прикладом неспрямованого пошуку є розглянутий простий випадковий пошук

Спрямований випадковий пошук. У цьому випадку окремі випробування пов'язані між собою. Результати проведених випробувань використовуються

для формування наступних. Як правило, випадковість використовується при формуванні напрямку спуску. Збіжність таких методів, як правило, вище, але самі методи зазвичай призводять тільки до локальних екстремумів.

4.1. Алгоритми локального пошуку

4.1.1. Простий випадковий пошук

Розглядається задача пошуку екстремуму функції з обмеженнями. Необхідно вирішити задачу мінімізації функції $f(\bar{x})$ при умові, що $\bar{x} \in [A, B]$.

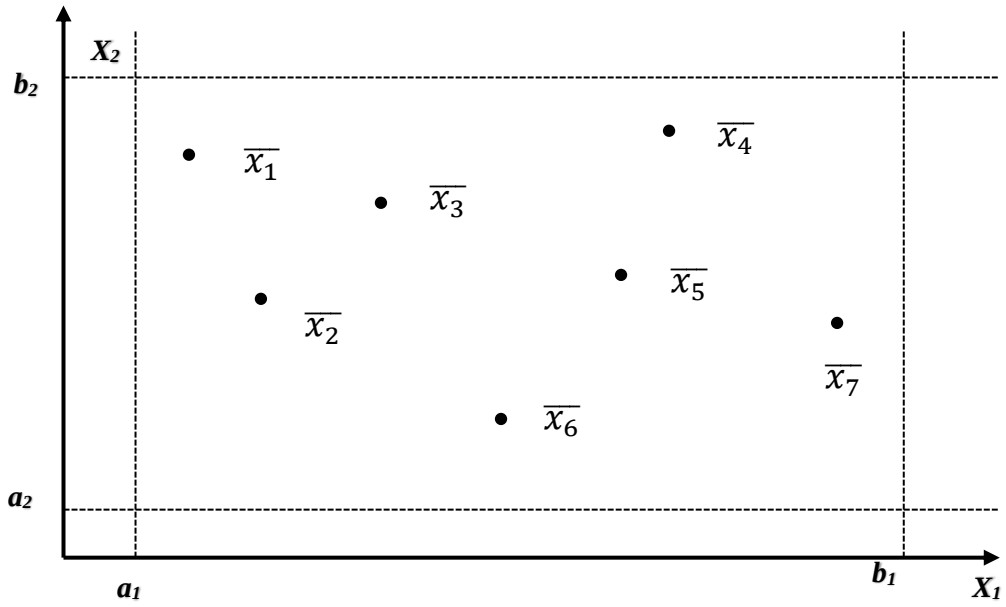


Рис. 4.1.1.1. Простий випадковий пошук

В даній області за рівномірним законом вибираємо випадкову точку $\overline{x_1}$ і обчислюємо в ній значення функції $y_1 = f(\overline{x_1})$. Потім таким же чином вибираємо випадкову точку $\overline{x_2}$ і обчислюємо $y_2 = f(\overline{x_2})$. Запам'ятовуємо мінімальне з цих значень і точку, в якій значення функції мінімальне. Далі генеруємо нову точку. Робимо N експериментів після чого кращу точку беремо в якості рішення задачі (точку в якій функція має мінімальне значення серед всіх «випадково» генерованих) [17; 20].

Оцінимо число експериментів необхідне для знаходження рішення (точки мінімуму) із заданою точністю. Нехай n - розмірність вектору змінних. Об'єм n-мірного прямокутника в якому ведеться пошук мінімуму,

$$V = \prod_{i=1}^{n}(b_i - a_i)$$

Якщо необхідно знайти рішення з точністю ε_1, $i = \overline{1,n}$, по кожній із змінних, то ми повинні потрапити в окіл оптимальної точки з об'ємом

$$V_\varepsilon = \prod_{i=1}^{n}\varepsilon_i$$

Вірогідність влучення в цей окіл при одному випробуванні дорівнює $P_\varepsilon = \frac{V_\varepsilon}{V}$. Ймовірність не потрапляння дорівнює $1 - P_\varepsilon$. Випробування незалежні, тому ймовірність не потрапляння за N експериментів дорівнює $(1 - P_\varepsilon)^N$. Ймовірність того, що ми знайдемо рішення за N випробувань:

$$P = 1 - (1 - P_\varepsilon)^N$$

З цього рівняння ми легко можемо отримати оцінку необхідного числа випробувань для відповідного мінімуму з відповідною точністю:

$$N \geq \frac{\ln(1 - P)}{\ln(1 - P_\varepsilon)}$$

Спираючись на задану точність ε_1, $i = \overline{1,n}$, і величину V, можна виявити P_ε та, задаючись ймовірністю P, подивитися як змінюється необхідна кількість експериментів N в залежності від P_ε і P (див. табл. 4.1.1.1).

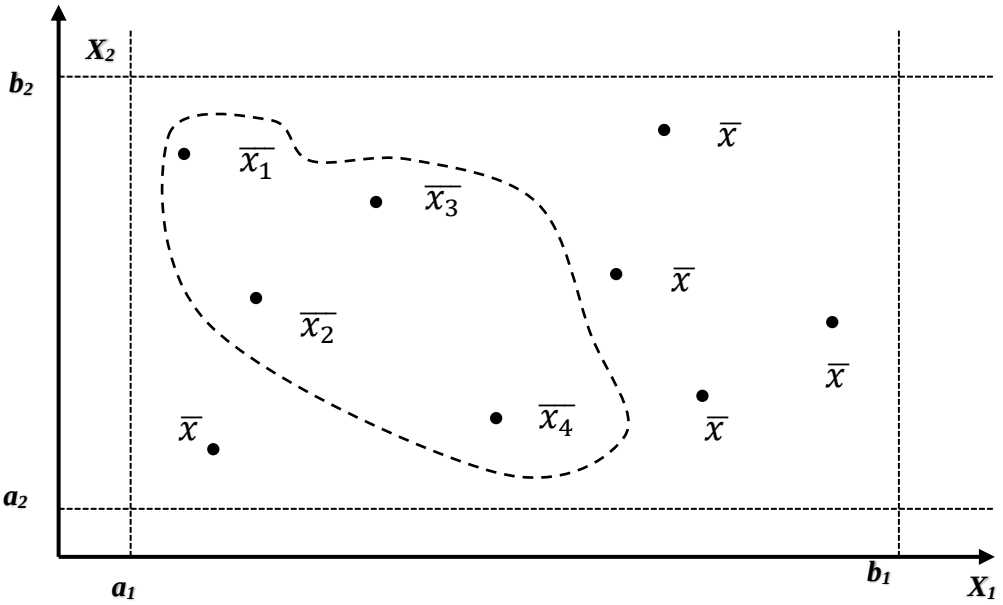


Рис. 4.1.1.2. Простий випадковий пошук на складній області

Для екстремальних задач на областях зі складною геометрією зазвичай записують цю область в n-мірний паралелепіпед. Для цього генерують в n-мірному паралелепіпеді випадкові точки по рівномірному закону, залишаючи тільки ті, які попадають в задану допустиму область [19].

Таблиця 4.1.1.1. Необхідна кількість експериментів.

P_ε	P				
	0.8	0.9	0.95	0.99	0.999
0.1	16	22	29	44	66
0.025	64	91	119	182	273
0.01	161	230	299	459	688
0.005	322	460	598	919	1379
0.001	1609	2302	2995	4603	6905

4.1.2. Алгоритм парної проби

В алгоритмі парної проби розділені пробний і робочий кроки [18; 20].

Нехай $\bar{x}^k$ – знайдено на k-му кроці найменше значення мінімізуємої функції $f(\bar{x})$. По рівномірному закону генерується випадковий одиничний вектор $\bar{\xi}$ і по обидві сторони від вихідної точки $\bar{x}^k$ робляться дві проби, тобто

проводимо обчислення функції у точка $\bar{x}_{1,2}^k = \bar{x}^k \pm g * \bar{\xi}$, де g - величина пробного кроку.

Робочий крок робиться у напрямі найменшого значення цільової функції. Чергове наближення визначається співвідношенням

$$\bar{x}^{k+1} = \bar{x}^k + \Delta\bar{x}^k = \bar{x}^k + a * \bar{\xi} * sign\left[f\left(\bar{x}^k - g\bar{\xi}\right) - f(\bar{x}^k + g\bar{\xi})\right]$$

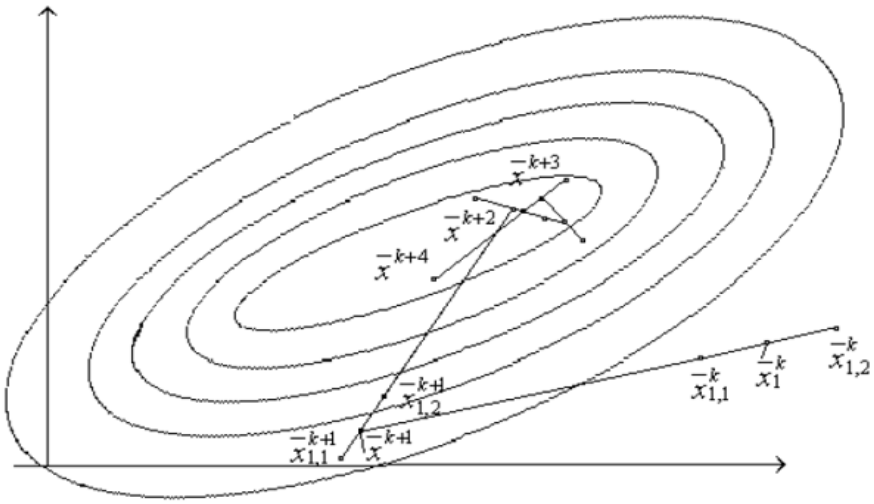


Рис. 4.1.2.1. Алгоритм парної проби

Особливістю цього алгоритму є його здатність до «блукання». Навіть коли знайшли екстремум, алгоритм здатен процес пошуку направити в інший бік.

4.1.3. Алгоритм найкращої проби

На поточному k-му кроці ми маємо точку $\bar{x}^k$. Генерується m випадкових одиничних векторів $\bar{\xi}_1$,..., $\bar{\xi}_m$ Робляться пробні кроки у напрямках

$g * \bar{\xi}_1,\ldots,g * \bar{\xi}_m$

і в точках

$\bar{x}^k + g * \bar{\xi}_1,\ldots, \bar{x}^k + g * \bar{\xi}_m$

вираховуються значення функції. Обирається той напрямок, який призводить до найбільшого зменшення функції:

$$\bar{\xi}^* = argmin\, f(\bar{x}^k + g * \bar{\xi}_i), \qquad i = \overline{1,m}.$$

У даному напрямку робиться крок

$\Delta\bar{x}^k = \lambda * \bar{\xi}^*$

Параметр λ може визначатися як результат мінімізації у напрямку, який визначається найкращою пробою, або вибиратися за певним правилом [18; 20].

Із збільшенням числа проб обраний напрямок наближається до напряму $-\nabla f(\bar{x})$

Якщо функція $f(\bar{x})$ близька до лінійної, то є можливість прискорити пошук, роздивляючись разом із найкращою і найгіршу пробу. Тоді робочий крок можна робити або у напрямку найкращої, або у напрямку, протилежному найгіршій пробі.

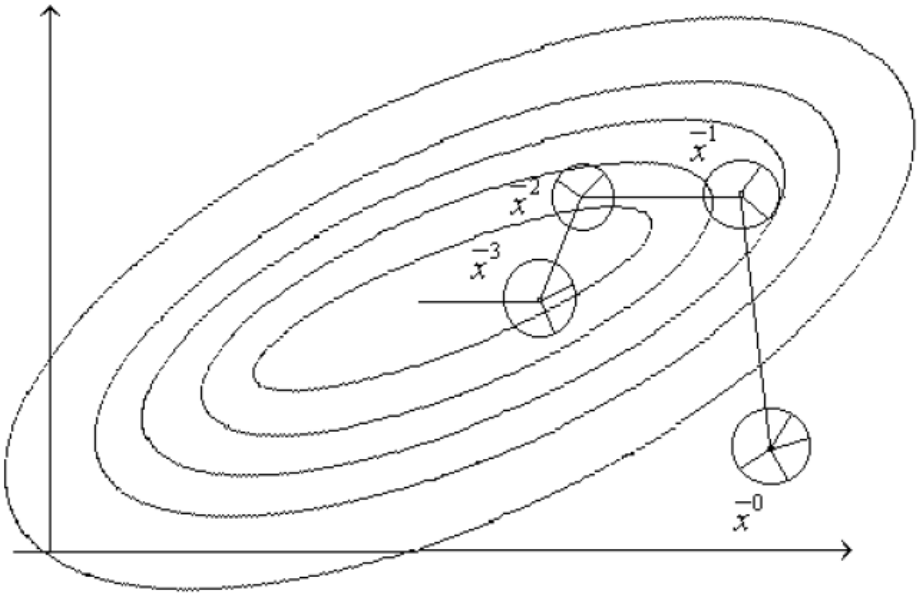


Рис. 4.1.3.1. Алгоритм найкращої проби

4.1.4. Метод статичного градієнту

Це один з найбільш простих та ефективних методів випадкового пошуку [17].

З початкової точки $\bar{x}^k$ в m випадкових напрямах робиться m незалежних проб

$g * \bar{\xi}_1, \dots, g * \bar{\xi}_m,$

а потім обчислюються значення мінімізуючої функції у відповідних точках. Для кожної проби запам’ятовуємо приріст функції

$$\Delta f_j = f\left(\bar{x}^k + g * \bar{\xi}_j\right) - f\ (\bar{x}^k).$$

Після цього формуємо векторну суму

$$\Delta \bar{f} = \sum_{j=1}^{m} \bar{\xi}_j * \ \Delta f_j$$

В межі при $m \to \infty$ напрям $\Delta \bar{f}$ співпадає з напрямом градієнта цільової функції. При скінченному m вектор $\Delta \bar{f}$ являє собою статичну оцінку напряму градієнта. В напрямі $\Delta \bar{f}$ робиться робочий крок. В результаті наступне приближення визначається відношенням

$$\bar{x}^{k+1} = \bar{x}^k - \ \lambda * \frac{\Delta \bar{f}}{||\Delta \bar{f}||}$$

При виборі оптимального значення λ, яке мінімізує функцію в заданому напрямі, ми отримуємо статистичний варіант методу найшвидшого спуску. Суттєва перевага перед детермінованими алгоритмами полягає в можливості прийняття рішення про напрям робочого кроку при m < n. При m = n и невипадкових ортогональних робочих кроках, напрямлених вздовж осей координат, алгоритм вироджується в градієнтний метод.

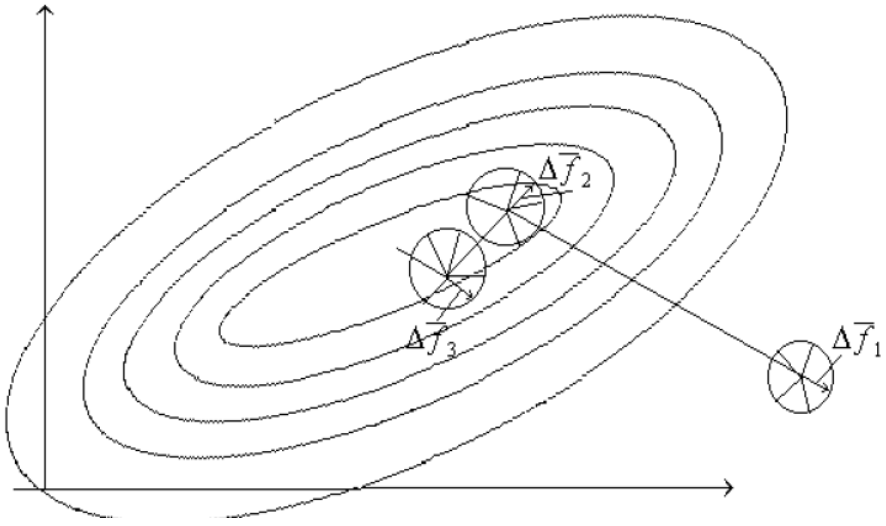


Рис. 4.1.4.1. Метод статичного градієнту

4.1.5. Алгоритм найкращої проби з направляючим гіперкубом

Всередині допустимої області будується гіперкуб [20]. В ньому випадково розташовується m точок $\bar{x}_1, \ldots, \bar{x}_m$, в яких вираховується значення

функції. Серед побудованих точок обираємо найкращу. Таким чином, на 1-му етапі координати випадкових точок задовольняють нерівності

$$a_i^1 \le x_i \le b_i^1,\ i = \overline{1,n},$$

$$\bar{x}^1 = \arg \min_{j=\overline{1,m}} \{f(\bar{x}_j)\}$$

– точка з мінімальним значенням цільової функції.

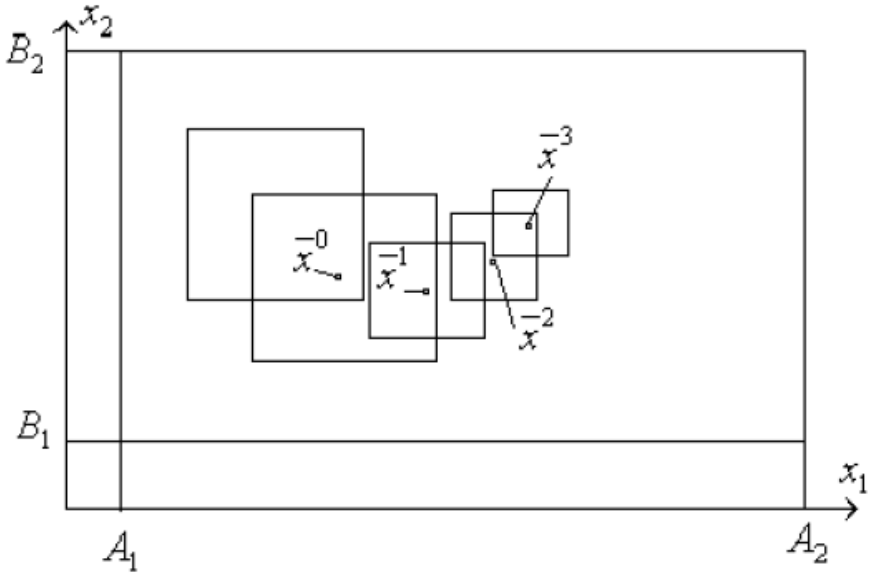


Рис. 4.1.5.1. Алгоритм найкращої проби з направляючим гіперкубом

Спираючись на цю точку, будуємо новий гіперкуб. Точка, в якій досягається мінімум функції на k-му етапі, вважається центром нового гіперкуба на (k+1)-му етапі.

Координати вершин гіперкубу на (k+1)-му етапі визначаються відношеннями

$$a_i^{k+1} = x_i^{k+1} - \frac{b_i^k - a_i^k}{2},\ b_i^{k+1} = x_i^{k+1} + \frac{b_i^k - a_i^k}{2},$$

де $\bar{x}^k$ – найкраща точка в гіперквадраті на k-му етапі.

В новому гіперкубі виконуємо ту саму послідовність дій, розташовуючи випадковим чином m точок. В результаті відбувається напрямлене переміщення гіперкуба в сторону зменшення функції.

В алгоритмі з навчанням, сторони гіперкуба можуть регулюватися згідно зі змінами по де-якому правилу параметра α, який визначає стратегію зміни

сторони гіперкуба. В цьому випадку координати вершин гіперкуба на (k+1)-му етапі будуть визначатись відношеннями

$$a_i^{k+1} = x_i^{k+1} - \frac{b_i^k - a_i^k}{2\alpha},\ b_i^{k+1} = x_i^{k+1} + \frac{b_i^k - a_i^k}{2\alpha}.$$

Добре відібране правило регулювання сторони гіперкубу приводить до достатньо ефективному алгоритму пошуку.

В алгоритмах випадкового пошуку, замість направляючого гіперкубу можуть використовуватись направляючі гіперсфери.

4.2. Алгоритми глобального пошуку

Випадковий пошук набуває вирішального значення при рішенні багатьох екстремальних задач і оптимізації складних об'єктів. В загальному випадку рішення багатьох екстремальних задач майже неможливе без елементу випадковості.

Розглянемо де-які варіанти пошуку глобального екстремуму [18; 20].

Варіант 1. В допустимій області D випадковим чином обирають точку $\bar{x}_1 \in D$. Прийнявши цю точку за початкову та використовуючи де-який детермінований метод чи алгоритм направленого випадкового пошуку, виконується спуск в точку локального мінімуму $\bar{x}_1^* \in D$, в області тяжіння якого опинилась точка $\bar{x}_1$.

Далі обирається нова випадкова точка $\bar{x}_2 \in D$, і таким же чином виконується спуск в точку локального мінімуму $\bar{x}_2^* \in D$ і т.д.

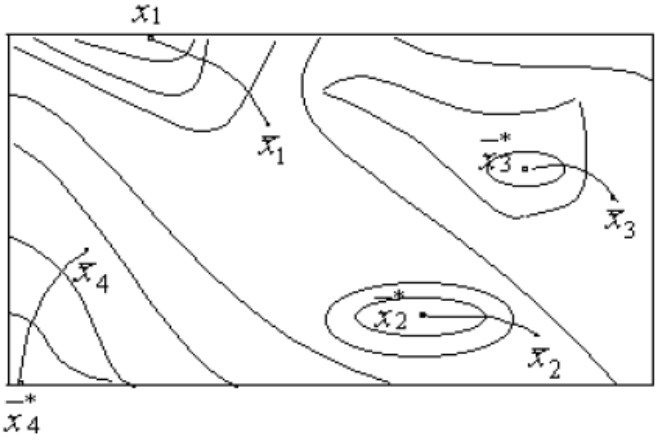


Рис. 4.2.1. Варіант 1

Пошук припиняється, як тільки за задане число m раз не вдається знайти точку локального екстремуму зі значенням функції, меншим попередніх.

Варіант 2. Нехай отримана де-яка точка локального екстремуму $\bar{x}_1^* \in D$. Після цього переходимо до ненаправленого випадкового пошуку до отримання точки $\bar{x}_2$, $f(\bar{x}_2) < f(\bar{x}_1^*)$.

З точки $\bar{x}_2$ за допомогою детермінованого алгоритму чи направленого випадкового пошуку отримаємо точку локального екстремума $\bar{x}_2^*$, в якій виконується нерівність $f(\bar{x}_2^*) < f(\bar{x}_1^*)$.

Далі за допомогою випадкового пошуку визначаємо нову точку $\bar{x}_3$, для якої справедлива нерівність $f(\bar{x}_3) < f(\bar{x}_2^*)$, і знову спуск у точку локального екстремума $\bar{x}_3^*$, і т.д.

Пошук припиняється, якщо при генерації де-якого числа нових випадкових точок не вдається знайти кращу, ніж попередній локальний екстремум, який тоді і використовується в якості рішення.

Варіант 3. Нехай $\bar{x}_1^0$ – де-яка початкова точка пошуку в області D, з якої виконується спуск в точку локального екстремуму $\bar{x}_1^*$ зі значенням $f(\bar{x}_1^*)$. Далі з точки $\bar{x}_1^*$ рухаємось або у випадковому напрямі, або у напрямі $\bar{x}_1^* - \bar{x}_1^0$ доти, доки функція знову не почне зменшуватись (виходимо з області тяжіння $\bar{x}_1^*$).

Отримана точка $\bar{x}_2^0$ приймається за початок наступного спуска. В результаті знаходимо новий локальний екстремум $\bar{x}_2^*$ зі значенням функції $f(\bar{x}_2^*)$.

Якщо $f(\bar{x}_2^*) < f(\bar{x}_1^*)$, точка $\bar{x}_1^*$ забувається і її місце займає точка $\bar{x}_2^*$. Якщо $f(\bar{x}_2^*) > f(\bar{x}_1^*)$, то повертаємось в точку $\bar{x}_1^*$ і рухаємось з неї у випадковому напрямку.

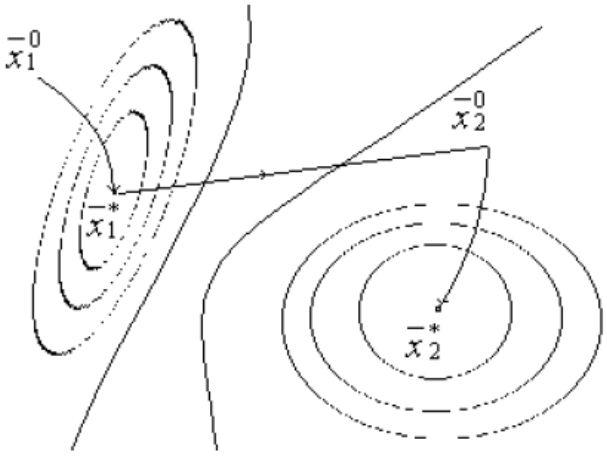


Рис. 4.2.2. Варіант 3

Процес зупиняється, якщо не вдається знайти кращий локальний мінімум після заданої кількості спроб або не вдається знайти «випадкового» напряму, в якому функція починає спадати.

Такий підхід дозволяє знайти глобальний екстремум у випадку зв’язних допустимих областей.

Варіант 4. В допустимій області D розташовуємо m випадкових точок і обираємо з них найкращу, тобто ту, в якій значення функції є мінімальним. З обраної точки виконуємо локальний спуск. А далі навколо траєкторії спуску утворюємо заборонену область.

В залишеній області випадковим чином розташовуємо точки і з кращої з них виконуємо спуск в точку локального екстремуму. Навколо нової траєкторій також будуємо заборонену область і так далі.

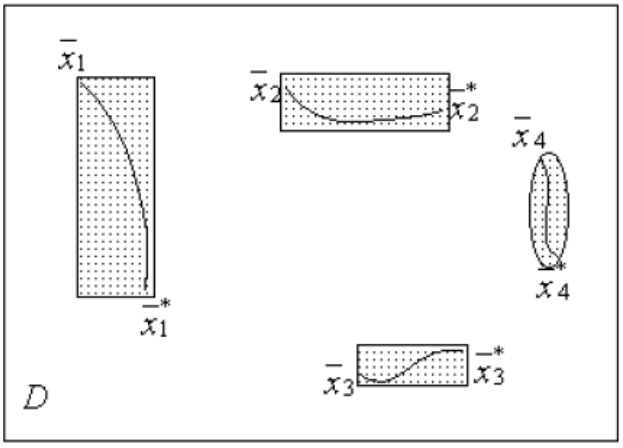


Рис. 4.2.3. Варіант 4

Пошук зупиняється, якщо не вдається знайти найкращого локального екстремуму за задану кількість спроб.

Зауваження: Комбінація випадкового пошуку з детермінованими методами застосовується не тільки для рішення задач з багатьма екстремумами. Часто до такої комбінації звертаються в ситуаціях, коли детерміновані методи стикаються з тими чи іншими причинами (застрягають на дні вузького яру, у сідловій точці, тощо). Крок у випадковому напрямку часом дозволяє долати таку ситуацію, тупикову для детермінованого алгоритму.

Питання до розділу 4.

1. Простий випадковий пошук?
2. Направлений випадковий пошук и ненаправлений? В чому різниця?
3. Приклади направленого випадкового пошуку?
4. Приклади ненаправленого випадкового пошуку?
5. Алгоритм метода статистичних градієнтів?
6. Приклади побудови алгоритмів глобального пошуку?

Перелік використаних джерел

1. Жалдак М. І., Гриус Ю. В. Основи теорії і методів оптимізації. Черкаси: Брама Україна. — 2005. — 608 с.
2. Аоки М. Введение в методы оптимизации. М.: Наука, 1977. – 344с.
3. Банди, В. Методы оптимизации. Вводный курс / В. Банди ; пер. С англ. – М. : Радио и связь, 1988. – 128 с.
4. Лавров Є. А. Математичні методи дослідження операцій : підручник / Є. А. Лавров, Л. П. Перхун, В. В. Шендрик та ін. – Суми : Сумський державний університет, 2017. – 212 с. ISBN 978-966-657-730-9
5. Есипов, Б. А. Методы исследования операций [Электронный ресурс] : учебное пособие / Б. А. Есипов. – СПб. : Изд-во «Лань», 2010. – URL : http://e.lanbook.com/books/pdf.php?book_id=144&p_id=25&booki=87
6. Шукаев Д.Н. Прикладные методы оптимизации: учебник. – М.: Издательский дом Академии Естествознания, 2017. – 212 с.

7. Алексеева Е. В., Кутненко О. А., Плясунов А. В. Численные методы оптимизации: Учеб. пособие / Новосиб. ун-т. Новосибирск, 2008. 128 с.
8. Григорків В.С. Оптимізаційні методи та моделі : підручник / В.С. Григорків, М.В. Григорків. – Чернівці : Чернівецький нац. ун-т, 2016. – 400 с.
9. В. П. Северин. Методы одномерного поиска : учебно-метод. пособ. по курсу «Методы оптимизации» / В. П. Северин. – Х. : НТУ «ХПИ», 2012. – 112 с. – ISBN 978-966-5930973-3
10. Попова Т.М. Методы безусловной оптимизации : Тексты лекций. / Т. М. Попова; [науч. ред. Р. В. Намм]. - Хабаровск: Изд-во Тихоокеан. гос. ун-та, 2013. – 76 с. ISBN 987-5-7389-1245-0
11. Кононюк А.Е. Основы теории оптимизации. Безусловная оптимизация К.2.ч.1. Киев:"Освіта України", 2011. - 544 с. ISBN 978-966-7599-50-8
12. Ахмадиев Ф.Г., Гильфанов Р.М. Математическое моделирование и методы оптимизации: Учебное пособие / Ф.Г. Ахмадиев, Р.М. Гильфанов. – Казань: Изд-во Казанск. гос. архитект.-строит. ун-та, 2017. – 178 с. Аттетков, А. В. Методы оптимизации: Учебное пособие / А.В. Аттетков, В.С. Зарубин, А.Н. Канатников. - М.: ИЦ РИОР: НИЦ Инфра-М, 2019. - 270 с.: ил.; - (Высшее образование: Бакалавриат). - ISBN 978-5-16-103309-8. - Текст : электронный. - URL: https://znanium.com/catalog/product/1002733
13. О. К. Молодід. Чисельні методи нелінійного програмування : методичні вказівки / НТУУ «КПІ» ; уклад.– Електронні текстові дані. – Київ : НТУУ «КПІ», 2015. – 44 с https://ela.kpi.ua/handle/123456789/11635 Лесин В. В. Основы методов оптимизации : учебное пособие / В. В. Лесин, Ю. П. Лисовец. — 4-е изд., стер. — Санкт-Петербург : Лань, 2016. — 344 с. — ISBN 978-5-8114-1217-4. — // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/86017
14. Лук'яненко С.О. Числові методи в інформатиці: Навч. посіб. – К.: "Видавництво "Політехнік", 2007. – 140с.
15. Растригин Л.А. Случайный поиск. Издательство «Знание». Москва, 1979.
16. Растригин Л.А. Случайный поиск в задачах оптимизации многопараметрических систем. Рига, Зинатне,1965. 212 с.
17. Монахов О.И., Корнеев П.А. Поисковые алгоритмы оптимизации в решении задач теории автоматического управления. Учебно-методическое пособие к практическим занятиям и курсовому проекту по дисциплине «Методы оптимизации». -М .: МГУПС (МИИТ), 2017. - 49 с.
18. Матренин П.В. Методы стохастической оптимизации: учебное пособие /П.В. Матренин, М.Г. Гриф, В.Г. Секаев. – Новосибирск: Изд-во НГТУ, 2016. – 67 с. ISBN 978-5-7782-2861-0