НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

Факультет прикладної математики

Кафедра прикладної математики

Курсова робота

із дисципліни «Методи оптимізації»

на тему: «Метод найшвидшого спуску»

|  |  |
| --- | --- |
| Студента групи КМ-03  Передерея Б. О. | Керівник:  Старший викладач Ладогубець Т. С.  Кількість балів:\_\_\_\_\_\_\_ |
|  | Оцінка:\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

Київ – 2022

**Зміст**

[**Постановка задачі** 2](#_Toc136334574)

[**Теоретична частина** 2](#_Toc136334575)

[**Основна частина** 3](#_Toc136334576)

[**Безумовна оптимізація** 3](#_Toc136334577)

[**Список використаної літератури** 13](#_Toc136334578)

# **Постановка задачі**

Дослідити збіжність методу найшвидшого спуску при мінімізації функції Розенброка в залежності від:

1. Величини кроку h при обчисленні похідних.
2. Схеми обчислення похідних.
3. Способу обчислення кроку: постійний, оптимальний.
4. Виду методу одновимірного пошуку (ДСК-Пауелла або Золотого перетину).
5. Точності методу одновимірного пошуку.
6. Значення параметру в алгоритмі Свена.
7. Вигляду критерію закінчення. .
8. Наявності модифікацій (методи Бута, Люстерніка, важкої кульки).

Використати метод штрафних функцій (метод зовнішньої точки) для умовної оптимізації при розташування локального мінімума поза випуклої допустимої області.

# **Теоретична частина**

Один з найпростіших методів першого порядку оптимізації функцій - метод найшвидшого спуску. Цей метод базується на знаходженні мінімуму функції шляхом здійснення кроків у напрямку, протилежному градієнту функції, тобто у напрямку найшвидшого спуску, осклільки від’ємний градієнт у точці направлений у строну найбільшого зменшення по всім значенням і він є ортогональним лінії рівня у точці .

Алгоритм методу найшвидшого спуску можна описати наступним чином. Спочатку задавши початкову точка x0, проводиться ітераційний процес, на кожному кроці якого виконується наступне:

1. Обчислюється градієнт функції в точці :
2. Знаходиться напрямок спуску, який дорівнює протилежному градієнту з нормуванням:

при сталому кроці або при оптимальному

1. Виконується визначення кроку , який мінімізує функцію (він може бути як сталим, так і оптимальним).
2. Обчислюється нова точка як:
3. Якщо задана точність не досягнута, повторюється ітераційний процес.

Від’єдним градієнт дає лише направлення оптимізації, але не велечину кроку. При цьому можна використовувати різні методи та алгоритми отримання оптимального (різні методи одновимірного пошуку для мінімізації цільової функції від ) та сталого значення кроку (наприклад, алгоритм Adam).

Хоча метод є простим у реалізації, він дуже низьку швидкість збіжності, що обгрунтовується як теоретичною частиною методу, так і практичними дослідами, тому на практиці для мінімазції цільових функцій він використовується дуже рідко. Зачасту метод найшвидого спуску використовують у випадках, коли інші методи використати неможливо, або коли функція має дуже велику кількість параметрів (наприклад як у нейронних мережах), оскільки оновлення самих параметрів функції здійснюється поступово.

Для пришвидшення збіжності методу можуть використовуватися його модифікації, такі як метод Бута, Люстерніка та важкої кульки.

Метод найшвидшого спуску часто використовується як частину інших методів оптимізації, наприклад, у методі Флетчера-Рівса або в інших методах в якості першого кроку.

Щодо умовної оптимізації, у цій курсовій роботі буде розглянуто метод штрафних функцій. Цей метод може бути представленим у різних варіаціях, які, однак, мають одну спільну рису – у всіх цих методах задача нелійного програмування перетворюється або на одну (еквівалентну початковій) задачу без обмежень, або в еквіваленту послідовність задач без обмежень. Перевага, яку ми отримуємо за рахунок переходу від задачі мінімізації за наявності обмежень до завдання мінімізації у відсутність обмежень, у тому, що в останньому випадку мінімізація може здійснюватися за допомогою набагато простіших (порівняно з першим випадком) алгоритмів (у цій курсовій роботі – методом найшвидшого спуску). При використанні методів штрафних функцій виходить максимальний оптимізаційний ефект за рахунок постійного компромісу між необхідністю задоволення обмежень та процесом мінімізації функції, який досягається шляхом присвоєння належних вагів функціям, що задають обмеження.

Одним з параметричних методів штрафних функцій є метод зовнішньої точки. Цей метод генерує послідовність точок, які виходять за межі допустимої області, аде дають рішення у допустимій області. Сама штрафна функція не дозволяє вектору x занадто сильно відійти від границі допустимої області, оскільки у цьому методі використовуються квадратичні штрафи з певним коефіцієнтом R у випадку, якщо обмеження не виконується, і прирівнює квадратичний штраф до нуля, якщо воно виконується.

Підсумовючи, основою методів штрафних функцій в області нелінійного програмування покладена ідея перетворення загальної нелінійної задачі у послідовність задач без обмежень шляхом додавання до цільової функції однієї чи декількох функцій, що задають обмеження, з тим, щоб обмеження, як такі, у задачі оптимізації не фігурували.

# **Основна частина**

## **Безумовна оптимізація**

*Вплив величини кроку h при обчисленні похідних*

Початкові умови:

Початкова точка: (-1.2, 0)

Критерій закінчення:

Величина похибки: 0.001

МОП: Золотий переріз

Величина похибки МОП: 0.001

Величина параметру в алгоритмі Свена: 0.01

Схема похідної: центральна

Дельта лямбда у Свені: 0.01 \*

Результати:

|  |  |  |  |
| --- | --- | --- | --- |
| Величина кроку h | Точка мінімуму, до якої прийшов алгоритм | Значення у точці мінімуму | Кількість обчислень функції |
| 0.1 | [0.32693 0.10388] | 0.45392 | 843 |
| 0.01 | [0.76475 0.58438] | 0.05536 | 3166 |
| 0.001 | [0.81534 0.66392] | 0.03417 | 2408 |
|  | [0.76806 0.58935] | 0.05383 | 2751 |
|  | [0.83631 0.69889] | 0.02682 | 1640 |
|  | [0.76914 0.59021] | 0.05348 | 2239 |
|  | [0.92755 0.85999] | 0.00526 | 1699 |
|  | [0.78929 0.6225 ] | 0.04442 | 2558 |
|  | [0.77247 0.59617] | 0.0518 | 2219 |
|  | [0.8144 0.66232] | 0.03453 | 2588 |
|  | [0.77646 0.60246] | 0.04999 | 3138 |
|  | [0.8317 0.69126] | 0.02835 | 2587 |
|  | [0.78355 0.6136 ] | 0.04686 | 1730 |
|  | [0.80381 0.64559] | 0.03852 | 849 |
|  | [0.9663 0.93346] | 0.00114 | 1591 |

Величина кроку впливала на результат нелінійно, тобто зменшення величини кроку не гарантували підвищення точності. З наведеної таблиці найкраща себе показали , яке підходить більше для зменшення кількості обчислень функції, але з трохи гіршими результатами, або , яка має на 742 обчислень більше, ніж , але це h дало у 30 раз більшу точність.

*Вплив схем обчислення похідних*

Початкові умови:

Початкова точка: (-1.2, 0)

Критерій закінчення:

Величина похибки: 0.001

МОП: Золотий переріз

Величина похибки МОП: 0.001

Величина параметру в алгоритмі Свена: 0.01

Величина кроку у похідних: h =

Дельта лямбда у Свені: 0.01 \*

Результати:

|  |  |  |  |
| --- | --- | --- | --- |
| Схема похідних | Точка мінімуму, до якої прийшов алгоритм | Значення у точці мінімуму | Кількість обчислень функції |
| Центральна | [0.9663 0.93346] | 0.00114 | 1591 |
| Лівостороння | [0.81311 0.66078] | 0.03494 | 1417 |
| Правостороння | [0.90273 0.81438] | 0.00949 | 1513 |

Изображение выглядит как линия, диаграмма, График, оригами

Автоматически созданное описание

*Графік кількості обчислень функції залежно від значення h та схеми похідних*

Изображение выглядит как снимок экрана, линия, График, диаграмма

Автоматически созданное описание

*Графік значення функції залежно від значення h та схеми похідних*

Згідно графіку кількості обчислень функцію найкраще себе показала правостороння похідна при точності з кількістю обчислень функції 507, але з графіку значень отримане мінімальне значення функції є дорівнює 0.4086. Найкращі з отриманих результатів:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Похідна | h | Точка мінімуму, до якої прийшов алгоритм | Значення у точці мінімуму | Кількість обчислень функції |
| Лівостороння |  | [0.90474 0.81790] | 0.009117 | 1165 |
| Правостороння |  | [0.95204 0.90578] | 0.00234 | 1380 |
| Центральна |  | [0.96630 0.93346] | 0.00114 | 1591 |

На основі наведеної таблиці можна обрати потрібний результат у залежності від того, що нам важливіше – кількість обчислень або точність отриманих результатів. Хоча між центральною та лівосторонньою границею різниця у кількості обчислень функції 426, точність центральної схеми більше у 8 разів.

*Способу обчислення кроку: постійний, оптимальний.*

Початкова точка: (-1.2, 0)

Критерій закінчення:

Величина похибки: 0.0001

МОП: Золотий переріз

Величина похибки МОП: 0.001

Величина параметру в алгоритмі Свена: 1

Величина кроку у похідних: h =

Схема похідних: Центральна

Дельта лямбда у Свені: 1 \*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Крок | Зменшення лямбда при збільшені функції | Точка мінімуму, до якої прийшов алгоритм | Значення у точці мінімуму | Кількість обчислень функції |
|  | У 2 рази | [0.99917 0.99834] | 0.0 | 33104 |
|  | У 2 рази | [0.99019 0.98047] | 0.0001 | 15526 |
|  | У 2 рази | [1.00144 1.00282] | 0.0 | 55 |

Найкраще себе показала , але цей результат є дуже унікальним, оскільки результати дуже залежать від інших параметрів методу та від початкової точки. Наприклад, почнемо з точки (-1.1, 0) з тими самими параметрами алгоритму. Результати:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Крок | Зменшення лямбда при збільшені функції | Точка мінімуму, до якої прийшов алгоритм | Значення у точці мінімуму | Кількість обчислень функції |
|  | У 2 рази | [0.98224 0.96476] | 0.00032 | 14075 |
|  | У 2 рази | [0.99019 0.98046] | 0.0001 | 14546 |
|  | У 2 рази | [1.00063 1.00126] | 0.0 | 25775 |

Тобто підібрані параметри для 55 обчислень при дуже сильно залежать від початкової точки і не зберігають своїх властивостей при найменшій зміні. Щодо роботи лямбди оптимальної:

Початкова точка: (-1.2, 0)

|  |  |  |  |
| --- | --- | --- | --- |
| Крок | Точка мінімуму, до якої прийшов алгоритм | Значення у точці мінімуму | Кількість обчислень функції |
| МОП (золотий переріз) | [0.96776 0.93647] | 0.00104 | 100 |
| МОП (дск) | [0.95041 0.90334] | 0.00246 | 100 |

Початкова точка: (-1.1, 0)

|  |  |  |  |
| --- | --- | --- | --- |
| Крок | Точка мінімуму, до якої прийшов алгоритм | Значення у точці мінімуму | Кількість обчислень функції |
| МОП (золотий переріз) | [1.07562 1.15734] | 0.00573 | 7863 |
| МОП (дск) | [0.96295 0.9308 ] | 0.00261 | 84 |

Як виявилося, оптимальний крок є більш стійким до зміни початкових параметрів і робить як мінімум у два рази менше обчислень цільової функції, аніж константний крок. Це можна пояснити тим, що фіксований крок не використовує жодної інформації про саму функцію, окрім факту зменшення функції у новій точці. Можливо, результати були би кращими, якби було використано отримання кроку за допомогою інформацію про значення похідних чи функцій на минулих кроках (наприклад алгоритм Adam, який використовує інформацію про значення функції на минулих кроках і є популярним алгоритмом у нейронних мережах), але у цій роботі інші (більш кращі методи) для отримання константної лямбди досліджуватися не будуть.

*Вплив виду методу одновимірного пошуку та точності методу одновимірного пошуку*

Початкова точка: (-1.2, 0)

Критерій закінчення:

Величина похибки: 0.001

Величина параметру в алгоритмі Свена: 0.01

Величина кроку у похідних: h =

Дельта лямбда у Свені: 0.01 \*

Результати:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Метод МОП | Точність МОП | Точка мінімуму, до якої прийшов алгоритм | Значення у точці мінімуму | Кількість обчислень функції |
| Золотий перетин | 1 | [-0.5701 0.33418] | 2.47361623 | 35 |
| Золотий перетин |  | [-0.5701 0.33418] | 2.47361623 | 35 |
| Золотий перетин |  | [ 0.4266 0.18053] | 0.32900201 | 829 |
| Золотий перетин |  | [ 0.77247 0.59617] | 0.05179839 | 2219 |
| Золотий перетин |  | [ 0.97698 0.95444] | 0.00053006 | 7898 |
| Золотий перетин |  | [ 0.99979 0.99958] | 4e-08 | 14138 |
| Золотий перетин |  | [ 0.99939 0.99878] | 3.7e-07 | 270146 |
| Золотий перетин |  | [ 0.99994 0.99988] | 0.0 | 463535 |
| Золотий перетин |  | [ 0.99998 0.99997] | 0.0 | 600053 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Метод МОП | Точність МОП | Точка мінімуму, до якої прийшов алгоритм | Значення у точці мінімуму | Кількість обчислень функції |
| ДСК Пауелла | 1 | [-0.51361 0.26943] | 2.29419939 | 40 |
| ДСК Пауелла |  | [-0.51361 0.26943] | 2.29419939 | 40 |
| ДСК Пауелла |  | [ 0.67567 0.45514] | 0.10538051 | 482 |
| ДСК Пауелла |  | [ 0.91256 0.83244] | 0.00765649 | 3304 |
| ДСК Пауелла |  | [ 0.99458 0.98917] | 2.941e-05 | 4248 |
| ДСК Пауелла |  | [ 0.99914 0.99829] | 7.3e-07 | 21956 |
| ДСК Пауелла |  | [ 0.9609 0.92327] | 0.00152913 | 28523 |
| ДСК Пауелла |  | [ 0.99994 0.99988] | 0.0 | 166113 |
| ДСК Пауелла |  | [ 0.99999 0.99999] | 0.0 | 216985 |

Що метод золотого перетину, що метод ДСК Пауелла, хоч і дали майже повністю мінімізували цільову функцію, але виконували велику кількість кроків. При точності від до ДСК Пауелла було у 1.5-2 рази гірше за метод золотого перерізу згідно кількості обчислень функції, але при більшій величині похибки методу найшвидшого спуску (окрім ) ДСК Пауелла виконувало у 1.5-3 рази менше обчислень цільової функції при похибці.

*Значення параметру в алгоритмі Свена*

Початкова точка: (-1.2, 0)

Критерій закінчення:

МОП: Золотий переріз

Величина похибки МОП: 0.001

Величина похибки: 0.001

Величина кроку у похідних: h =

Схема похідної: центральна

|  |  |  |  |
| --- | --- | --- | --- |
| Параметр Свена | Точка мінімуму, до якої прийшов алгоритм | Значення у точці мінімуму | Кількість обчислень функції |
| 10 | [0.96943 0.93934] | 0.00096 | 135 |
| 1 | [0.9682 0.93747] | 0.00101 | 95 |
| 0.1 | [0.86142 0.74089] | 0.01934 | 1433 |
| 0.01 | [0.9663 0.93346] | 0.00114 | 1591 |
| 0.001 | [0.80349 0.64522] | 0.03863 | 1019 |
|  | [0.81851 0.66870] | 0.0331 | 2585 |
|  | [0.79936 0.63767] | 0.04043 | 3595 |
|  | [0.86143 0.7412 ] | 0.01927 | 3552 |
|  | [0.99189 0.98379] | 7e-05 | 2543 |
|  | [0.78858 0.62151] | 0.04471 | 2386 |
|  | [0.77267 0.59568] | 0.05186 | 2985 |
|  | [0.88787 0.78798] | 0.01258 | 164 |
|  | [0.94176 0.88671] | 0.0034 | 2360 |
|  | [0.82962 0.68779] | 0.02905 | 5008 |
|  | [0.77881 0.60602] | 0.04895 | 4336 |
|  | [0.78053 0.60765] | 0.04841 | 4042 |
|  | [0.76276 0.58028] | 0.05651 | 9157 |

Закономірність між зменшенням параметру в алгоритмі Свена та кількістю обчислень цільової функції немає лінійної залежності. Найкраще себе показали значення 1 та . Що цікаво – при значенню 1 через дуже великий крок у Свені, сам алгоритм дуже часто повертав інтервал [-1; 1], підраховуючи цільову функцію лише два рази (в обох точках функція зростала). Значення у Свені заставляло видавати значення з більшою оптимізацією. Але швидкі тести показали, що збільшення загальної точності МНС дуже швидко нівелює ефект значення параметру Свена. При параметру у Свені функція виконує 8030 обчислень функції. Але значення потрібно дослідити, варіюючи інші гіперпараметри

*Наявності модифікацій (методи Бута, Люстерніка)*

Початкова точка: (-1.2, 0)

Критерій закінчення:

Величина похибки: від до

МОП: Золотий переріз

Величина похибки МОП: 0.0001

Величина параметру в алгоритмі Свена: 1

Величина кроку у похідних: h =

Схема похідних: Центральна

Дельта лямбда у Свені: 1 \*

Без модифікації (оптимальний крок):

|  |  |  |  |
| --- | --- | --- | --- |
| Величина похибки | Точка мінімуму, до якої прийшов алгоритм | Значення у точці мінімуму | Кількість обчислень функції |
|  | [0.944811 0.892425] | 0.0030517290534505083 | 59 |
|  | [0.944811 0.892425] | 0.0030517290534505083 | 59 |
|  | [0.944811 0.892425] | 0.0030517290534505083 | 59 |
|  | [0.9549 0.911692] | 0.0020360036153700685 | 2992 |
|  | [0.997625 0.995244] | 5.654671001116072e-06 | 11150 |
|  | [1.000018 1.000036] | 3.187467096938542e-10 | 71341 |
|  | [1.000017 1.000035] | 2.9935832718279436e-10 | 71446 |
|  | [0.999996 0.999992] | 1.653632936741718e-11 | 186229 |
|  | [1. 1.] | 4.845172548228179e-15 | 352854 |
|  | [1. 1.] | 3.454693824482231e-17 | 354345 |

Метод Бута

|  |  |  |  |
| --- | --- | --- | --- |
| Величина похибки | Точка мінімуму, до якої прийшов алгоритм | Значення у точці мінімуму | Кількість обчислень функції |
|  | [0.848868 0.722703] | 0.023292838671245996 | 73 |
|  | [0.850066 0.722198] | 0.022497284225225064 | 104 |
|  | [0.850712 0.722537] | 0.022424769012345636 | 139 |
|  | [0.960754 0.922862] | 0.0015437126434409516 | 2653 |
|  | [0.993329 0.986682] | 4.454622335755933e-05 | 25126 |
|  | [1.000229 1.000458] | 5.238276656513157e-08 | 36056 |
|  | [0.999976 0.999952] | 5.8495937052521e-10 | 127090 |
|  | [0.999995 0.99999 ] | 2.3596883061668478e-11 | 128483 |

Без модифікації (сталий крок)

|  |  |  |  |
| --- | --- | --- | --- |
| Величина похибки | Точка мінімуму, до якої прийшов алгоритм | Значення у точці мінімуму | Кількість обчислень функції |
|  | [1.000966 1.003067] | 0.000129444104671098 | 36 |
|  | [1.000966 1.003067] | 0.000129444104671098 | 36 |
|  | [1.000966 1.003067] | 0.000129444104671098 | 36 |
|  | [1.001442 1.002824] | 2.4594578419536843e-06 | 55 |
|  | [1.001415 1.002834] | 2.0023628766815063e-06 | 68 |
|  | [1.000002 1.000005] | 5.8375053802681465e-12 | 37763 |
|  | [1.000002 1.000005] | 5.8375053802681465e-12 | 37763 |
|  | [1.000002 1.000005] | 5.8375053802681465e-12 | 37763 |

Метод Люстерніка

/ДОРОБИТИ/

## **Умовна оптимізація**

# **Список використаної літератури**

1. Himmelblau D. M. Applied nonlinear programming. New York : McGraw-Hill, 1972. 498 p. (page 72-83)
2. C M. J. Steepest Descent. OSTI.GOV | U.S. Department of Energy Office of Scientific and Technical Information. URL: https://www.osti.gov/servlets/purl/983240 (date of access: 29.05.2023).
3. The Method of Steepest Ascent (Descent). Department of Mathematical Sciences | Montana State University. URL: https://math.montana.edu/jobo/st578/sec6.pdf (дата звернення: 29.05.2023).