**Лабораторна робота №4**

**Тема: „*Наближені методи одновимірної оптимізації для унімодальних функцій* ”**

***Мета:*** розібрати методи та реалізувати алгоритми оптимізації унімодальних функцій.

**Питання.**

1. Унімодальні функції та їх властивості.
2. Методи пошуку мінімуму для унімодальних функцій:
3. метод дихотомії;
4. метод золотого перерізу;
5. метод Фібоначчі;
6. метод парабол.

**І. Теоретичні відомості.**

Необхідність дослідження задач одновимірної оптимізації та розробка чисельних методів їх розв'язування обумовлена, по-перше, тим, що цей клас екстремальних задач є зручною моделлю для розробки і теоретичного дослідження ефективності методів багатовимірної оптимізації, а, по-друге, тим, що в багатьох методах відшукання екстремуму таких багатовимірних задач використовується одновимірний пошук.

Класичні методи одновимірної мінімізації мають досить обмежені застосування, оскільки обчислення похідних в практичних задачах не завжди можливе. Наприклад, значення функції  визначаються в результаті спостережень чи деякого фізичного експерименту, і отримати інформацію про її похідну важко або взагалі неможливо в силу недиференційовності функції, що оптимізується. Але навіть у тих випадках, коли похідну все ж таки вдається обчислити, відшукання коренів рівняння  і визначення інших точок, підозрілих на екстремум, може бути пов'язане з серйозними труднощами. Тому важливо мати такі методи пошуку екстремуму, які не вимагають обчислення похідних і є більш зручними для реалізації на ЕОМ.

Якщо про цільову функцію  відомо лише те, що вона є неперервною на заданій множині , то для знаходження точки глобального мінімуму, взагалі кажучи, необхідно досліджувати цю функцію в усіх точках множини . Але оскільки нас цікавить відшукання деякого наближення точки мінімуму, то для цього функцію досліджують у скінченній кількості точок. Різні способи вибору такої сукупності точок визначають і різні методи одновимірної мінімізації.

В математиці розроблено багато ефективних методів для розв'язування задач одновимірної мінімізації, орієнтованих на різні класи функцій, що зустрічаються при розв'язуванні прикладних задач.

Більшість наближених методів одновимірної мінімізації у процесі своєї реалізації генерують послідовність відрізків , що стягується до однієї з точок . Однак, гарантувати приналежність точки  відрізку  для деякого  можна лише для певного класу функцій, що мінімізуються на множині .

Далі розглянемо кілька найбільш поширених методів одновимірної мінімізації на класі функцій, у яких всі точки локального мінімуму є точками глобального мінімуму.

**1. Унімодальні функції та їх властивості**

Нехай функція  визначена і скінченна на множині .

**Означення 1.** Функція  називається *унімодальною* на відрізку , якщо вона неперервна на  і якщо існують точки  і  такі, що

1) якщо , то на відрізку   строго монотонно спадає;

2) якщо , то на відрізку   строго монотонно зростає;

3) якщо , то .

На рис. 1 наведено приклади унімодальних функцій, а на рис. 2 – ні. Згідно означення 1 відрізки ,  є відрізками монотонності, а відрізок  є відрізком сталості функції . Як видно з рис. 1, для унімодальних функцій можливі випадки, коли один або два з відрізків  вироджуються в точку.

З означення 1 випливає, що для унімодальних функцій точки локального мінімуму є точками глобального мінімуму, при цьому множина точок мінімуму  (рис. 1). Очевидно, що опуклі на проміжку  функції є унімодальними на цьому проміжку.

Якщо в означенні 1 , то функція  називається  *строго унімодальною* на відрізку  (рис. 1 а), г)).

З означення строго унімодальної функції випливає, що така функція не містить ділянок сталості і в неї єдина точка локального мінімуму. Очевидно, що строго і сильно опуклі функції є строго унімодальними.

Для класу унімодальних функцій мають місце наступні твердження.

**Лема 1.** *Якщо функція*  *унімодальна на відрізку**, то вона унімодальна і на відрізку* .

**Лема 2.** *Нехай функція*  *унімодальна на відрізку*  *і* *. Тоді*

1)*якщо* *то* *, а якщо*  *то* ;

2) *якщо*  *то* *, тобто відрізок*  *містить хоча б одну точку* *.*

**Наслідок.** *Якщо*  *то існує точка*  *така, що* *, а якщо*  *то* .

 

a) б)

 

в) г)

Рис. 1.

 

а) б)

Рис. 2.

На практиці для перевірки унімодальності диференційованих на відрізку  функцій використовують наступні твердження.

**Лема 3.** *Якщо функція*  *диференційована на*  *і*  *не спадає при всіх* *, то*  *– унімодальна*.

**Лема 4**. *Якщо*  *– двічі диференційована функція на*  *і*  *при всіх* *, то*  *– унімодальна*.

Приклад 1. Довести, що функція



унімодальна на відрізку .

Оскільки функція  двічі диференційована, то знайдемо спочатку першу похідну

,

а потім другу

.

Оскільки  і  для всіх , то  – унімодальна на цьому відрізку згідно леми 4.

**Зауваження.** Леми 3 і 4 є достатніми умовами опуклості диференційованих і двічі диференційованих функцій відповідно (див. наслідки теорем 1 і 6)

Методи мінімізації унімодальних функцій та аналіз їх ефективності відіграють значну роль, оскільки такі функції часто зустрічаються при розв'язуванні практичних задач. Крім того, припущення про унімодальність функції в околі точки мінімуму  є, взагалі кажучи, досить природним (див. рис. 25.2). Тому відшукання такого околу, який називається відрізком локалізації точки мінімуму, є важливим попереднім етапом розв’язування задачі одновимірної мінімізації.

**Алгоритм методу дихотомії**

Нехай задано ,  і відрізок , де функція  унімодальна. Покласти.

# **Крок 1.** Знайти точки

# ,

# і обчислити , .

**Крок 2.** Якщо , то покласти , , інакше , , .

**Крок 3.** Якщо , то покласти  і перейти на крок 4, в протилежному випадку покласти  і перейти до виконання кроку 1.

**Крок 4.** Вивести , . Кінець алгоритму.

Оскільки кожен крок методу вимагає обчислення значення функції  у двох точках, то для досягнення потрібної точності  необхідно зробити  обчислень значень функції, при цьому з (3) маємо

. (4)

Звідси випливає, що число кроків алгоритму задовольняє умову

.

Часто на практиці число  можливих обчислень значень функції  задане заздалегідь і його перебільшення небажане.

В цьому випадку нерівність (4) дає можливість оцінити точність отриманого наближення  після  обчислень значень функції:

.

початок

f(x),a,b,eps

ϭ=eps/3

x1 = (a + b - delta) / 2; x2 = (a + b + delta) / 2;  
f1 = function(x1); f2 = function(x2);

f1 <= f2

b = x2

a = x1

*abs*(a-b) > *EPS*

x\*=(a + b) / 2; f\*=f(x\*)

x\*, f\*

кінець

Блок схема алгоритму дихотомії

**Алгоритм модифікованого методу золотого перерізу**

Нехай задано  і відрізок , на якому функція  унімодальна.

**Крок 1.** Знайти точки  і обчислити , .

**Крок 2**. Якщо , то покласти   , , , , знайти  і якщо , то обчислити ;

інакше () покласти , , , , , , знайти  і якщо , то обчислити .

**Крок 3.** Якщо , то перейти на виконання кроку 5.

**Крок 4.** Якщо , то перейти до виконання кроку 2, інакше перейти на виконання кроку 1.

**Крок 5**.Вивести . Кінець алгоритму.

початок

f(x),a,b,eps

u = a + (3 - Math.*sqrt*(5)) / 2 \* (b - a);v = a + b - u;fu = function(u);

fv = function(v);

fu <= fv

b = v; v = u; fv = fu;u = a + b - v; fu = function(u);

a = u; u = v; fu = fv; v = a + b - u;  
fv = function(v);

u > v

x\*=(a + b) / 2; f\*=f(x\*)

x\*, f\*

кінець

u = a + (3 - Math.*sqrt*(5)) / 2 \* (b - a); v = a + b - u;  
fu = function(u); fv = function(v);

Math.*abs*(b - a) > *EPSILON*

Блок схема алгоритму золотого перерізу

**Алгоритм модифікованого методу Фібоначчі**

Нехай задано  – число обчислень значень унімодальної функції  i відрізок .

**Крок 0.** Покласти .

**Крок 1.** Знайти точки

, 

і обчислити , .

**Крок 2.** Якщо , то покласти   , , , , знайти  і якщо , то обчислити ;

інакше () покласти , , , , , , знайти  і якщо , то обчислити .

**Крок 3.** Якщо , то перейти на виконання кроку 5.

**Крок 4.** Покласти . Якщо , то перейти до виконання кроку 2, інакше перейти на виконання кроку 1.

**Крок 5.** Вивести . Кінець алгоритму.

початок

f(x),a,b,eps

u = a + (3 - Math.*sqrt*(5)) / 2 \* (b - a);v = a + b - u;fu = function(u);

fv = function(v);

fu <= fv

b = v; v = u; fv = fu;u = a + b - v; fu = function(u);

a = u; u = v; fu = fv; v = a + b - u;  
fv = function(v);

u > v

x\*=(a + b) / 2; f\*=f(x\*)

x\*, f\*

кінець

u = a + (3 - Math.*sqrt*(5)) / 2 \* (b - a); v = a + b - u;  
fu = function(u); fv = function(v);

I=1..n

Блок схема алгоритму Фібоначі

**Алгоритм методу парабол**

Нехай задані відрізок  локалізації точки мінімуму функції  і  (досить мале число).

**Крок 0.** На відрізку  визначити першу вдалу трійку  і обчислити , , .

**Крок 1**. Знайти точку



і обчислити .

**Крок 2.** Якщо , то покласти ,  i кінець алгоритму, інакше якщо , то покласти , , , , інакше покласти , , , .

**Крок 3.** Для одержаних точок  знайти вдалу трійку , обчислити значення  і перейти до виконання кроку 1.

Кінець алгоритму.

початок

f(x),a,h,eps,x1

x1 = a - 0.5 \* h \* (function(a + h) - function(a - h)) / (function(a + h) - 2 \* function(a) + function(a - h));

*abs*(x1 - a) > *EPSILON*

res<=0

a+=0.1

a = x1;  
x1 = a - 0.5 \* h \* (function(a + h) - function(a - h)) / (function(a + h) - 2 \* function(a) + function(a - h));

x\*=(a + b) / 2; f\*=f(x\*)

x\*, f\*

кінець

**Індивідуальне завдання**

За допомогою програм для ЕОМ, які реалізують методи дихотомії, золотого перерізу, Фібоначчі і парабол, знайти наближений розв'язок задачі одновимірної мінімізації функції  на відрізку  з точністю , , . Для кожного методу визначити кількість зроблених кроків, кількість обчислень значень функції , а одержані результати подати у вигляді таблиці і порівняти їх:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **№** | Назва методу |  | | | |  |  | | | |  |
| **N*k*** | **N*f*** | ***x*\*** | ***f*\*** | **час** | **N*k*** | **N*f*** | ***x*\*** | ***f*\*** | **час** |
| **1.** | Дихотомії | 29 | 59 | -1.1463 | -6.4047 | 4045мс | 29 | 59 | -1.14635570 | -6.40474488 | 2911мс |
| **2.** | Золотого перерізу | 21 | 24 | -1.1463 | -6.4047 | 1454мс | 40 | 45 | -1.14635570 | -6.40474488 | 1588мс |
| **3.** | Фібоначчі | 15 | 18 | -1.1465 | -6.4047 | 3693мс | 15 | 18 | -1.14652473 | -6.40474422 | 3358мс |
| **4.** | Парабол | 5 | 46 | -1.1463 | -6.4047 | 571мс | 7 | 62 | -1.14635504 | -6.40474488 | 601мс |

**Вихідні дані з методу дихотомії**

х1 = -1,0017 x2 = -0,9983 f(x1) = -6,01 f(x2) = -5,99

х1 = -1,5008 x2 = -1,4975 f(x1) = -1,83 f(x2) = -1,94

х1 = -1,2512 x2 = -1,2479 f(x1) = -6,12 f(x2) = -6,13

х1 = -1,1265 x2 = -1,1231 f(x1) = -6,4 f(x2) = -6,39

х1 = -1,1889 x2 = -1,1855 f(x1) = -6,36 f(x2) = -6,37

х1 = -1,1577 x2 = -1,1543 f(x1) = -6,4 f(x2) = -6,4

х1 = -1,1421 x2 = -1,1387 f(x1) = -6,4 f(x2) = -6,4

х1 = -1,1499 x2 = -1,1465 f(x1) = -6,4 f(x2) = -6,4

х1 = -1,146 x2 = -1,1426 f(x1) = -6,4 f(x2) = -6,4

**Вихідні дані з методу золотого перерізу**

u = -1,5279 v = -1,2361 f(u) = -0,92 f(v) = -6,2

u = -1,2361 v = -1,0557 f(u) = -6,2 f(v) = -6,24

u = -1,0557 v = -0,9443 f(u) = -6,24 f(v) = -5,69

u = -1,1246 v = -1,0557 f(u) = -6,39 f(v) = -6,24

u = -1,1672 v = -1,1246 f(u) = -6,39 f(v) = -6,39

u = -1,1935 v = -1,1672 f(u) = -6,35 f(v) = -6,39

u = -1,1672 v = -1,1509 f(u) = -6,39 f(v) = -6,4

u = -1,1509 v = -1,1409 f(u) = -6,4 f(v) = -6,4

u = -1,1571 v = -1,1509 f(u) = -6,4 f(v) = -6,4

u = -1,1509 v = -1,1471 f(u) = -6,4 f(v) = -6,4

u = -1,1471 v = -1,1447 f(u) = -6,4 f(v) = -6,4

Вихідні дані з методу Фібоначчі

u = -1,2361 v = -1,0557 f(u) = -6,2 f(v) = -6,24

u = -1,0557 v = -0,9443 f(u) = -6,24 f(v) = -5,69

u = -1,1246 v = -1,0557 f(u) = -6,39 f(v) = -6,24

u = -1,1672 v = -1,1246 f(u) = -6,39 f(v) = -6,39

u = -1,1935 v = -1,1672 f(u) = -6,35 f(v) = -6,39

u = -1,1672 v = -1,1509 f(u) = -6,39 f(v) = -6,4

u = -1,1509 v = -1,1409 f(u) = -6,4 f(v) = -6,4

u = -1,1572 v = -1,1509 f(u) = -6,4 f(v) = -6,4

u = -1,1509 v = -1,1472 f(u) = -6,4 f(v) = -6,4

u = -1,1472 v = -1,1446 f(u) = -6,4 f(v) = -6,4

u = -1,1484 v = -1,1472 f(u) = -6,4 f(v) = -6,4

u = -1,1472 v = -1,1459 f(u) = -6,4 f(v) = -6,4

u = -1,1459 v = -1,1459 f(u) = -6,4 f(v) = -6,4

u = -1,1472 v = -1,1459 f(u) = -6,4 f(v) = -6,4

Вихідні дані з методу парабол

f(x+h) = 3,6239f(x-h) = 3,6239f(x) = 4

f(x+h) = -4,9332f(x-h) = -4,9332f(x) = -5

f(x+h) = -6,3052f(x-h) = -6,3052f(x) = -6

f(x+h) = -6,4039f(x-h) = -6,4039f(x) = -6

**Висновки**

Отже, реалізувавши методи знаходження екстремума функції на заданому проміжку, можна дійти висновку що найефективніший метод це парабол, тумо що йому потребується найменше часу на виконання.

**Додаток А**

**Метод дихотомії:**

var watch = System.Diagnostics.Stopwatch.StartNew();

count\_callFunc = 0;

iterCount = 0;

chart1.Series[1].Points.Clear();

double x\_min, x\_max;

double eps = E, d = eps / 3, x1, x2, f1, f2;

x\_min = Convert.ToDouble(numericUpDown1.Value);

x\_max = Convert.ToDouble(numericUpDown2.Value);

do

{

x1 = (x\_min + x\_max - d) / 2;

x2 = (x\_min + x\_max + d) / 2;

f1 = Function(x1);

f2 = Function(x2);

Console.WriteLine("х1 = " + x1 + " x2 = " + x2 + " f(x1) = " + f1 + " f(x2) = " + f2);

if (f1 <= f2)

{

x\_max = x2;

}

else

{

x\_min = x1;

}

iterCount++;

} while (Math.Abs(x\_max - x\_min) > eps);

time.Text = watch.Elapsed.TotalMilliseconds.ToString() + " мс";

watch.Stop();

double x\_res = (x\_min + x\_max) / 2, fx\_res = Function(x\_res);

chart1.Series[1].Points.AddXY(x\_res, fx\_res);

chart1.Series[1].Color = Color.Red;

x\_text.Text = "Мінімум " + fx\_res.ToString() + " 3";

x\_rres\_text.Text = "Значення Х: " + x\_res.ToString();

iter\_text.Text = "Кількість обчислень функцій " + count\_callFunc.ToString();

iter\_count.Text = "Кількість ітерацій " + iterCount.ToString();

chart1.Series[1].MarkerSize = 8;

**Метод золотого перерізу:**

var watch = System.Diagnostics.Stopwatch.StartNew();

count\_callFunc = 0;

iterCount = 0;

chart1.Series[1].Points.Clear();

double x\_min, x\_max, eps = E, u, v, fu, fv;

x\_min = Convert.ToDouble(numericUpDown1.Value);

x\_max = Convert.ToDouble(numericUpDown2.Value);

u = x\_min + (3 - Math.Sqrt(5)) / 2 \* (x\_max - x\_min);

v = x\_min + x\_max - u;

fu = Function(u);

fv = Function(v);

do

{

iterCount++;

if (fu <= fv)

{

x\_max = v;

v = u;

fv = fu;

u = x\_min + x\_max - v;

fu = Function(u);

}

else

{

x\_min = u;

u = v;

fu = fv;

v = x\_min + x\_max - u;

fv = Function(v);

}

if (u > v)

{

u = x\_min + (3 - Math.Sqrt(5)) / 2 \* (x\_max - x\_min);

v = x\_min + x\_max - u;

fu = Function(u);

fv = Function(v);

}

Console.WriteLine("u = " + u + " v = " + v + " f(u) = " + fu + " f(v) = " + fv);

} while (Math.Abs(x\_max - x\_min) > eps);

time.Text = watch.Elapsed.TotalMilliseconds.ToString() + " мс";

watch.Stop();

double x\_res = (x\_min + x\_max) / 2, fx\_res = Function(x\_res);

chart1.Series[1].Points.AddXY(x\_res, fx\_res);

chart1.Series[1].Color = Color.Red;

x\_text.Text = "Мінімум " + fx\_res.ToString() + " 3";

x\_rres\_text.Text = "Значення Х: " + x\_res.ToString();

iter\_text.Text = "Кількість обчислень функцій " + count\_callFunc.ToString();

iter\_count.Text = "Кількість ітерацій " + iterCount.ToString();

chart1.Series[1].MarkerSize = 8;

**Метод Фібоначі:**

var watch = System.Diagnostics.Stopwatch.StartNew();

count\_callFunc = 0;

iterCount = 0;

chart1.Series[1].Points.Clear();

double a, b, n = 15, u, v, fu, fv;

a = Convert.ToDouble(numericUpDown1.Value);

b = Convert.ToDouble(numericUpDown2.Value);

u = a + F(n) / F(n + 2) \* (b - a);

v = a + b - u;

fu = Function(u);

fv = Function(v);

for (int i = 0; i < n; i++)

{

if (fu <= fv)

{

b = v;

v = u;

fv = fu;

u = a + b - v;

fu = Function(u);

}

else

{

a = u;

u = v;

fu = fv;

v = a + b - u;

fv = Function(v);

}

if (u > v)

{

u = a + F(n - i + 1) / F(n - i + 3) \* (b - a);

v = a + b - u;

fu = Function(u);

fv = Function(v);

}

iterCount++;

Console.WriteLine("u = " + u + " v = " + v + " f(u) = " + fu + " f(v) = " + fv);

}

time.Text = watch.Elapsed.TotalMilliseconds.ToString() + " мс";

watch.Stop();

double x\_res = (a + b) / 2, fx\_res = Function(x\_res);

chart1.Series[1].Points.AddXY(x\_res, fx\_res);

chart1.Series[1].Color = Color.Red;

x\_text.Text = "Мінімум " + fx\_res.ToString() + " 4";

x\_rres\_text.Text = "Значення Х: " + x\_res.ToString();

iter\_text.Text = "Кількість обчислень функцій " + count\_callFunc.ToString();

iter\_count.Text = "Кількість ітерацій " + iterCount.ToString();

chart1.Series[1].MarkerSize = 8;

**Метод парабол:**

var watch = System.Diagnostics.Stopwatch.StartNew();

count\_callFunc = 0;

iterCount = 0;

chart1.Series[1].Points.Clear();

double h = 0.001;

double x, eps = E;

x = Convert.ToDouble(numericUpDown1.Value);

if (x == 0) x += 0.1;

//while ((Function(x + h) - 2 \* Function(x) + Function(x - h)) / (h \* h) <= 0)

//{

// x += 0.1;

// iterCount++;

//}

double x1;

x1 = x - 0.5 \* h \* (Function(x + h) - Function(x - h)) / (Function(x + h) - 2 \* Function(x) + Function(x - h));

while (Math.Abs(x1 - x) > eps)

{

x = x1;

x1 = x - 0.5 \* h \* (Function(x + h) - Function(x - h)) / (Function(x + h) - 2 \* Function(x) + Function(x - h));

iterCount++;

Console.WriteLine("f(x+h) = "+ Function(x + h)+"f(x-h) = " + Function(x + h) +"f(x) = " + Function(x));

}

time.Text = watch.Elapsed.TotalMilliseconds.ToString() + " мс";

watch.Stop();

double x\_res = x1, fx\_res = Function(x\_res);

chart1.Series[1].Points.AddXY(x\_res, fx\_res);

chart1.Series[1].Color = Color.Red;

x\_text.Text = "Мінімум " + fx\_res.ToString() + " 5";

x\_rres\_text.Text = "Значення Х: " + x\_res.ToString();

iter\_text.Text = "Кількість обчислень функцій " + count\_callFunc.ToString();

iter\_count.Text = "Кількість ітерацій " + iterCount.ToString();

chart1.Series[1].MarkerSize = 8;